Next-Generation Movie Recommendation Systems: A Hybrid Collaborative Filtering Approach

A Project Report submitted in the partial fulfillment of the Requirements

for the award of the degree

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted By

R. Gangadhar (21471A0521)
K. Akhil Kumar (21471A0532)
N. Gopi Krishna (21471A0539)

Under the esteemed guidance of

Dr. SIREESHA MOTURI, B.Tech, M.Tech., Ph.D.

Associate Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NARASARAOPETA ENGINEERING COLLEGE: NARASAROPET
(AUTONOMOUS)

Accredited by NAAC with A+ Grade and NBA under Tier -1 NIRF rank in the band of 251-320 and an ISO 9001:2015 Certified

Approved by AICTE, New Delhi, Permanently Affiliated to JNTUK, Kakinada

KOTAPPAKONDA ROAD, YALAMANDA VILLAGE, NARASARAOPET- 522601

2024-2025

NARASARAOPETA ENGINEERING COLLEGE: NARASAROPET
(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project work entitled **"MOVIE RECOMMENDATION SYSTEM"** is a Bonafide work done by the team R. Gangadhar (21471A0521), K. Akhil Kumar (21471A0532), N. Gopi Krishna (21471A0539) in partial fulfilment of the requirements for the award of BACHELOR OF TECHNOLOGY in the Department of COMPUTER SCIENCE AND ENGINEERING during 2024-2025.

PROJECT GUIDE	PROJECT CO-ORDINATOR
Dr. Sireesha Moturi, **B.Tech, M.Tech., Ph.D.**	**Dodda Venkata Reddy, B.Tech, M.Tech, (Ph.D).**
Associate Professor	**Assistant Professor**

HEAD OF THE DEPARTMENT	EXTERNAL EXAMINER
Dr. S. N. Tirumala Rao, M.Tech, Ph.D.	
Professor	

DECLARATION

We declare that this project work titled "MOVIE RECOMMENDATION SYSTEM" is composed by ourselves that the work contain here is our own except where explicitly stated otherwise in the text and that this work has not been submitted for any other degree or professional qualification except as specified.

By

R. Gangadhar	(21471A0521)
K. Akhil Kumar	(21471A0532)
N. Gopi Krishna	(21471A0539)

ACKNOWLEDGEMENT

We wish to express our thanks to carious personalities who are responsible for the completion of the project. We are extremely thankful to our beloved chairman sri **M. V. Koteswara Rao B.Sc.,** who took keen interest in us in every effort throughout this course. We owe out sincere gratitude to our beloved principal **Dr. S. Venkateswarlu, (Ph.D).,** for showing his kind attention and valuable guidance throughout the course.

We express our deep felt gratitude towards **Dr. S. N. Tirumala Rao M .Tech.,Ph.D,** Professor & Head of Department of CSE and also to our guide **Dr. Sireesha Moturi, B.Tech, M .Tech.,Ph.D,** Associate professor & Project coordinator of CSE department whose valuable guidance and unstinting encouragement enable us to accomplish our project successfully in time.

We extend our sincere thanks towards **Dodda Venkata Reddy, B.Tech, M .Tech.(Ph.D.),** Assistant professor & Project coordinator of the project for extending his encouragement. Their profound knowledge and willingness have been a constant source of inspiration for us throughout this project work.

We extend our sincere thanks to all other teaching and non-teaching staff to department for their cooperation and encouragement during our B.Tech degree. We have no words to acknowledge the warm affection, constant inspiration and encouragement that we received from our parents.

We affectionately acknowledge the encouragement received from our friends and those who involved in giving valuable suggestions had clarifying our doubts which had really helped us in successfully completing our project.

By

R. Gangadhar (21471A0521)
K. Akhil Kumar (21471A0532)
N. Gopi Krishna (21471A0539)

NEC NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

INSTITUTE VISION AND MISSION

INSTITUTION VISION

To emerge as a Centre of excellence in technical education with a blend of effective student centric teaching learning practices as well as research for the transformation of lives and community

INSTITUTION MISSION

M1: Provide the best class infra-structure to explore the field of engineering and research

M2: Build a passionate and a determined team of faculty with student centric teaching, imbibing experiential, innovative skills

M3: Imbibe lifelong learning skills, entrepreneurial skills and ethical values in students for addressing societal problems

NEC NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION OF THE DEPARTMENT

To become a center of excellence in nurturing the quality Computer Science & Engineering professionals embedded with software knowledge, aptitude for research and ethical values to cater to the needs of industry and society.

MISSION OF THE DEPARTMENT

The department of Computer Science and Engineering is committed to

M1: Mould the students to become Software Professionals, Researchers and Entrepreneurs by providing advanced laboratories.

M2: Impart high quality professional training to get expertise in modern software tools and technologies to cater to the real time requirements of the industry.

M3: Inculcate team work and lifelong learning among students with a sense of societal and ethical responsibilities.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Specific Outcomes (PSO's)

PSO1: Apply mathematical and scientific skills in numerous areas of Computer Science and Engineering to design and develop software-based systems.

PSO2: Acquaint module knowledge on emerging trends of the modern era in Computer Science and Engineering

PSO3: Promote novel applications that meet the needs of entrepreneur, environmental and social issues.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Educational Objectives (PEO’s)

The graduates of the programme are able to:

PEO1: Apply the knowledge of Mathematics, Science and Engineering fundamentals to identify and solve Computer Science and Engineering problems.

PEO2: Use various software tools and technologies to solve problems related to academia, industry and society.

PEO3: Work with ethical and moral values in the multi-disciplinary teams and can communicate effectively among team members with continuous learning.

PEO4: Pursue higher studies and develop their career in software industry.

NEC NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS)

Program Outcomes

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

NEC NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS)

Project Course Outcomes (CO'S):

CO421.1: Analyse the System of Examinations and identify the problem.

CO421.2: Identify and classify the requirements.

CO421.3: Review the Related Literature

CO421.4: Design and Modularize the project

CO421.5: Construct, Integrate, Test and Implement the Project.

CO421.6: Prepare the project Documentation and present the Report using appropriate method.

Course Outcomes – Program Outcomes mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C421.1		√											√		
C421.2	√		√		√								√		
C421.3				√		√	√	√					√		
C421.4			√			√	√	√					√	√	
C421.5					√	√	√	√	√	√	√	√	√	√	√
C421.6									√	√	√		√	√	

Course Outcomes – Program Outcome correlation

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C421.1	2	3											2		
C421.2			2		3								2		
C421.3				2		2	3	3					2		
C421.4			2			1	1	2					3	2	
C421.5					3	3	3	2	3	2	2	1	3	2	1
C421.6									3	2	1		2	3	

Note: The values in the above table represent the level of correlation between CO's and PO's

1. Low level
2. Medium Level
3. High level

Project mapping with various courses of Curriculum with Attained PO's:

Name of the course from which principles are applied in this project	**Description of the device**	**Attained PO**
C2204.2, C22L3.2	Gathering the requirements and defining the problem, plan to develop a movie recommendation system using machine learning methods like content, collaborative, and hybrid.	PO1, PO3
CC421.1, C2204.3, C22L3.2	Each and every requirement is critically analyzed, the process model is identified and divided into five key components: content-based filtering, collaborative filtering, hybrid methods, user interface, and data handling.	PO2, PO3
CC421.2, C2204.2, C22L3.3	Logical design is done by using the unified modelling language which involves individual team work.	PO3, PO5, PO9
CC421.3, C2204.3, C22L3.2	Each and every module is tested, integrated, and evaluated in our project.	PO1, PO5
CC421.4, C2204.4, C22L3.2	Documentation is done by all our four members in the form of a group.	PO10
CC421.5, C2204.2, C22L3.3	Each and every phase of the work in group is presented periodically.	PO10, PO11
C2202.2, C2203.3, C1206.3, C3204.3, C4110.2	Implementation is done and the project will be handled by the team; future updates in our project can be implemented based on user feedback and performance analysis.	PO4, PO7
C32SC4.3	The physical design includes hardware components like servers for data handling, and software components involving machine learning algorithms and database management.	PO5, PO6

ABSTRACT

In today's digital landscape, the overwhelming abundance of content presents users with decision fatigue when selecting entertainment options. Recommendation systems have emerged as critical tools to address this challenge, particularly in the movie streaming industry where personalization significantly enhances user experience and platform engagement.

This research presents an advanced hybrid approach to movie recommendation that integrates content-based filtering (CBF) and collaborative filtering (CF) methodologies to overcome the limitations inherent in each individual approach. The content-based component analyzes movie attributes including genres, descriptions, and taglines using TF-IDF vectorization and cosine similarity measures to identify thematically similar content. Simultaneously, the collaborative component employs Singular Value Decomposition (SVD) to process user-item interaction data, achieving an impressive RMSE of 0.68 and MAE of 0.89 in rating prediction accuracy.

Our proposed hybrid system implements a two-stage filtering process: pre-filtering to narrow the initial database using primary contextual factors, followed by post-filtering that incorporates additional contextual elements to refine recommendations. This approach is built upon the MovieLens-100k dataset, utilizing comprehensive movie metadata alongside user rating histories to generate personalized suggestions that balance content relevance with predicted user satisfaction.

The system addresses common challenges in recommendation systems, including the cold-start problem, by leveraging a weighted rating formula that considers both rating frequency and average scores to ensure recommendation quality and reliability. Performance evaluation demonstrates that our hybrid model outperforms standalone approaches, achieving an RMSE of 0.69 compared to the base publication's 0.76, indicating enhanced predictive capability. This research contributes to the evolving field of recommendation systems by demonstrating how sophisticated analytical methods and comprehensive datasets can be harnessed to deliver increasingly personalized and accurate content suggestions.

INDEX

S.NO.	**LIST OF FIGURES**	**PAGE NO**
1.	1.1: Example for Content-Based Recommendations	02
2.	1.2: Example for Collaborative Recommendation	03
3.	1.3: Example for Hybrid Recommendations	03
4.	1.4: Architecture diagram	04
5.	1.5: A workflow for Hybrid Movie Recommendation System	07
6.	3.1: Dataset before pre-processing	14
7.	3.2: Data Pre-processing	15
8.	3.3: Screenshot of removing null values	16
9.	3.4: Screenshot of cast attribute after pre-processing	16
10.	3.5: Screenshot of crew attribute after pre-processing	17
11.	3.6: Screenshot of genres attribute after pre-processing	17
12.	3.7 Dataset after pre-processing	18
13.	3.8: Movies recommended by implemented model	19
14.	3.9: Printing the similarity for every movie in the dataset	19
15.	3.10: RMSE scores of SVD model	20
16.	5.1: Hybrid Function	35
17.	5.2: Input movie name taken from the user	36
18.	5.3: Screenshots for Movie Recommendation System	37
19.	6.1: Output screen (1) for movie recommendation system	40
20.	6.2: Output screen (2) for movie recommendation system	40
21.	6.3: Recommending movies (1)	41

1.INTRODUCTION

1.1 Introduction

In our day-to-day life, we are going to use many applications on our mobile. For example, consider the application flip kart, when are searched for an item in that application it will show similar items to your search. Even if you close that particular application, it will send you a notification about similar items on your search again and again. Like this, there are many applications we are going to use in our daily life which recommends our favorite products. That means the recommendations are possible only when there is data about a user. Therefore, large organizations collect data from different users in multiple actions. Here, the data is stored in various formats. From that data, the organizations will recommend or predict what the user like most. How to recommend something? Mainly there are three ways to recommend something [5, 12].

Those are:

1. Content-Based Recommendations
2. Collaborative Recommendations
3. Hybrid Recommendations Here, let us go with one by one.

A. Content-Based Recommendations:

Content-Based Recommendations will be made recommendations based on the user search. For example, consider the content-based movie recommendation system, If a user searches for a particular movie, the system will show the results which are most similar to the search, or if a user liked a movie, Here also it will show similar movies to the liked movie. The data used to recommend a movie is dependent on the particular user. Data is nothing but information or raw data about a user or something. Recommendations will change from person to person [6, 9]. For better understanding about Content- Based Recommendation System consider Fig 1.1.

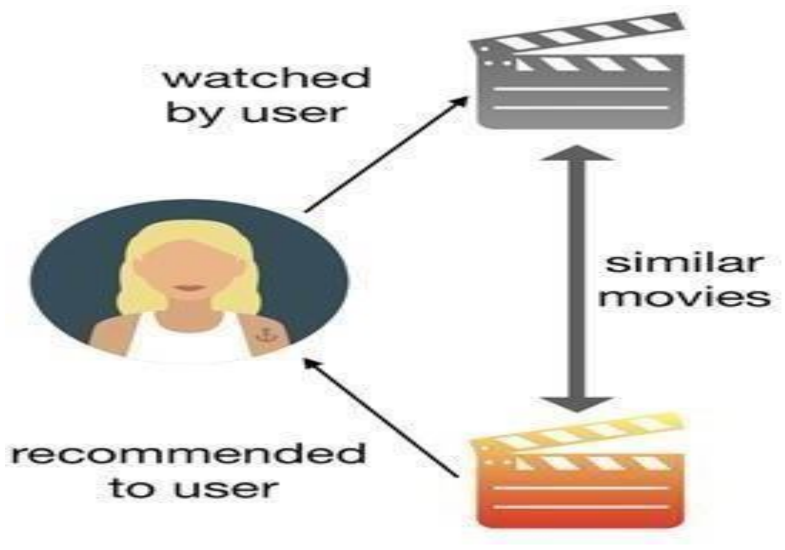


Fig 1.1: Example for Content-Based Recommendations

B. Collaborative Recommendations:

Collaborative Recommendations will be made recommendations based on similarity between the users either positively or negatively. Here, the data of multiple users are used to make recommendations. For example, consider the collaborative movie recommendation system, let us consider User-A, User-B, and User-C. User-A liked some movies, User-B liked some movies and User-C liked some movies. Then the movies which is not visited by User-A and which are liked by User-B will recommend to User-A and vice versa. If User-A-liked movies are almost all negative to User-C, therefore here it shows that the movies which are disliked by User-A are liked by User-C. Taking this into consideration the collaborative movie recommendation system will recommend the movies to the movies to User-A which are disliked by User-B. This is how the collaborative recommendations are done [7, 10]. For better understanding about Collaborative Recommendation System consider Fig 1.2.

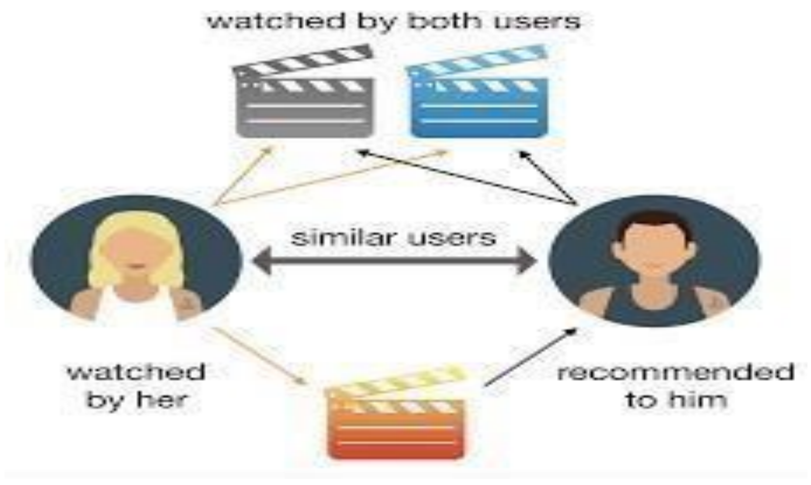


Fig 1.2: Example for Collaborative Recommendations

C. Hybrid Recommendations:

Both content-based recommendation system and collaborative recommendation systems have individual advantages and disadvantages to overcome the Hybrid Recommendation System introduced. Hybrid Recommendation is nothing but the combination of both advantages of a content-based recommendation system and a collaborative recommendation system [8, 11]. Fig 1.3 gives you the better explanation about how the Hybrid Recommendation System works.

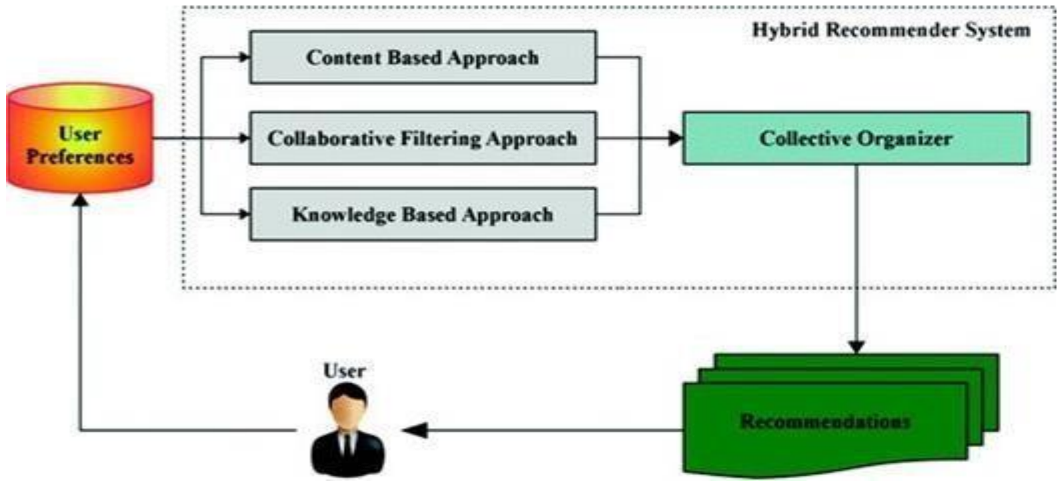


Fig 1.3: Example for Hybrid Recommendations

1.2 Existing System

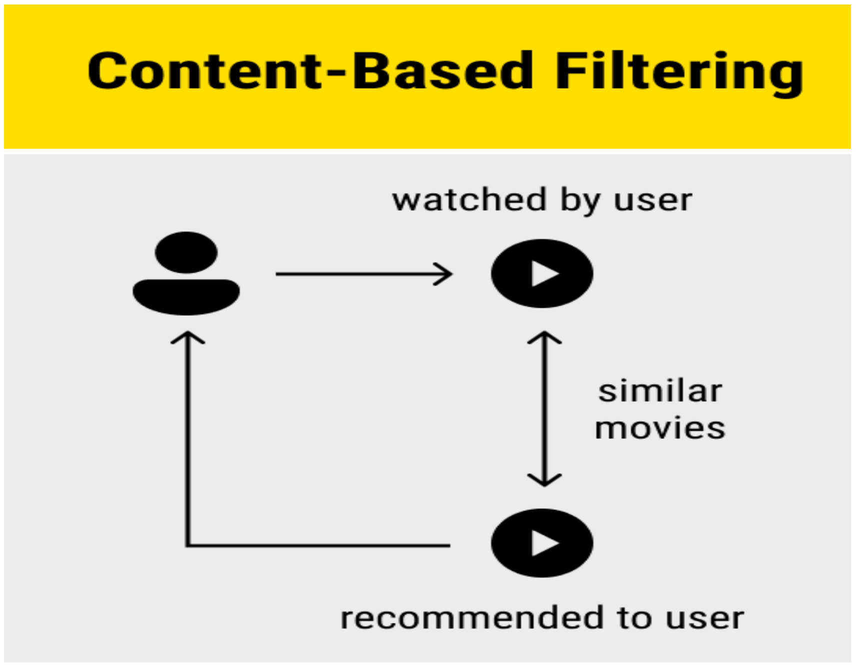


Fig 1.3: Architecture Diagram

Content-based movie recommendation system is a type of recommendation system that suggests movies to users based on the attributes of movies that the user has enjoyed in the past. The system analyzes the content of movies, such as genre, actors, directors, plot summary, and keywords, and creates a profile of the user's interests. The system then recommends movies to the user that are similar to the ones they have enjoyed in the past.

Content-based movie recommendation systems have several advantages over other types of recommendation systems. They provide personalized recommendations to users, are transparent and interpretable, and can handle new users or items without any prior data.

Content-based movie recommendation systems use different techniques to match the user's preferences with the attributes of movies. One popular technique is cosine similarity, which measures the similarity between two movies based on their attributes.

The system calculates the cosine similarity score between the movies and recommends the movies with the highest similarity score. Fig 1.4 gives the clear view of content-based filtering of movies.

Movie recommendation systems are designed to suggest movies to users based on their preferences and past movie choices. There are different types of recommendation systems, and here are some of the existing ones:

A. Collaborative Filtering: This type of recommendation system analyzes user behavior, such as movie ratings, and suggests similar movies to the user based on the ratings of other users who have similar tastes.

B. Content-Based Filtering: This type of recommendation system analyzes attributes of the movies, such as genre, actors, and directors, and suggests movies to the user based on their preferences.

C. Hybrid Filtering: This type of recommendation system combines collaborative and content- based filtering to provide personalized recommendations to the user.

D. Knowledge-Based Systems: This type of recommendation system asks the user a series of questions to determine their preferences and suggests movies based on their answers.

E. Demographic-Based Systems: This type of recommendation system suggests movies to users based on their age, gender, location, and other demographic factors.

F. Popularity-Based Systems: This type of recommendation system suggests popular movies to users based on their ratings and box office success.

G. Association Rules: This type of recommendation system suggests movies to users based on the association between movies. For example, if a user likes "The Lord of the Rings," the system may suggest "The Hobbit" or "Game of Thrones.

These are some of the existing movie recommendation systems. Each system has its advantages and disadvantages

Disadvantages:

1. Doesn’t generate accurate and efficient results.
2. It produces cost effective results

1.3 Proposed System

The proposed hybrid recommendation system is designed to leverage the strengths of both content- based and collaborative filtering methodologies to provide highly accurate and personalized suggestions. This system will first utilize content-based filtering to analyze item features, such as genres, tags, or descriptions, to create a profile of user preferences based on their interactions with similar items. Concurrently, the system will employ collaborative filtering techniques to tap into the wisdom of the crowd, examining user interaction patterns to detect preferences shared across a larger dataset of users. By integrating these approaches, the system aims to mitigate the limitations of each method—such as the cold start problem and limited scalability in content-based systems, and the sparsity issue in collaborative models. The fusion of these methods will be orchestrated through a weighted hybrid approach, where recommendations are dynamically adjusted based on the confidence level of predictions from each method. The outcome will be a more robust, scalable, and adaptive recommendation system that can cater to diverse user tastes and evolving content libraries.

Advantages of our hybrid recommendation system include:

1. Scalability: Our system can efficiently handle large datasets and provide real-time recommendations, thanks to the scalability of cosine similarity-based algorithms. Complex computations are minimized as the system primarily compares attribute similarities, rather than relying on intricate user behavior analysis.

2. Personalization: By leveraging past viewing history and preferences, our system delivers tailored suggestions to users. It analyzes attributes of movies the user has liked before and suggests similar ones based on cosine similarity scores, ensuring a personalized viewing experience.

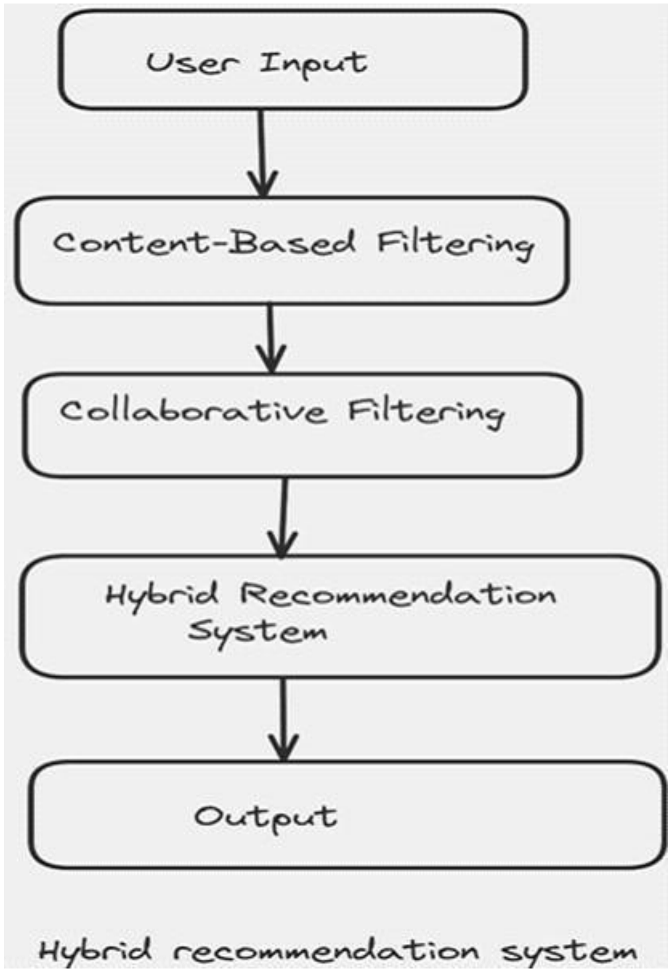


Fig 1.5: A workflow for Hybrid Movie Recommendation System

3. Transparency: Our recommendation system offers transparency by suggestions on the similarity between movie attributes rather than complex algorithms or user behavior patterns. Users can understand the rationale behind the recommendations, enhancing trust and satisfaction with the system.

4. Improved Recommendation Accuracy: By combining the strengths of collaborative filtering and content-based filtering, our hybrid system can provide more accurate and relevant movie recommendations to users. It leverages both user-item interactions and movie content features to ensure a comprehensive understanding of user preferences.

5. Addressing Cold Start Issues: The hybrid approach mitigates the cold start problem commonly encountered in recommendation systems, where new users or items lack sufficient interaction data for accurate recommendations. By incorporating content-based features, our system can still make relevant suggestions even for new users or items with limited interaction history.

6. Enhanced Personalization: Our hybrid recommendation system offers enhanced personalization by taking into account both explicit user preferences (collaborative filtering) and implicit preferences inferred from movie content (content-based filtering). This combination allows for a more tailored and individualized recommendation experience for each user.

7. Diverse Recommendation Coverage: The hybrid system can offer a wider range of movie recommendations by tapping into different recommendation strategies. Collaborative filtering may excel at suggesting popular or trending movies based on user similarities, while content-based filtering can recommend niche or less-known films based on specific content attributes.

1.3 System Requirements

1.3.1 Hardware Requirements:

• System type	:	intel®core™i5-7500UCPU@2.70gh
• Cache memory	:	4MB
• RAM	:	8 GB

1.3.2 Software Requirements:

• Operating system	:	windows 10, 64-bit OS
• Coding language	:	Python
• Python distribution	:	Anaconda, Spyder, Flask

2. RELATED WORK

2.1 Literature Survey

In order to improve recommendation accuracy, S. Agrawal et al. [1] proposed a hybrid strategy that combines content-based filtering and collaborative filtering techniques, employing Support Vector Machines (SVM) as a classifier. Their analysis, which used the Movie Lens dataset as a basis, showed a notable improvement above conventional technique. Similar comparisons were made between other recommender systems by N. Vaidya et al. [2], including keyword-based, hybrid, content-based, collaborative filtering, and demographic systems. They outlined the benefits and limitations of each strategy, stressing the value recommender systems in practical applications especially e-commerce platforms to increase revenue and enhance customer satisfaction.

Singular Value Decomposition (SVD) algorithm was used collaboratively by Y. Xiong et al. [4] to suggest photographs with comparable styles. To increase accuracy, they reduced dependencies in training data, optimized the recommendation algorithm, and employed binary ratings (1 for used, 0 for not used).

By classifying similar users according to preferences, S. Bhat et al. [5] solved the cold-start problem in their recommendation system by the use of a feature-based algorithm. In order to improve accuracy,J. Chen et al. [6] concentrated on making recommendations for tourist destinations based on reviews from actual travelers, taking into account elements like cost, lodging, and transportation.

P. Darshna et al. [7] investigated content-based and collaborative filtering for music recommendation, employing k-means clustering to analyze attributes like loudness and acoustic quality. In order to provide precise suggestions for college selection, S. Girase et al. [8] created a hybrid recommender that combines collaborative and content-based filtering. M.M. Reddy et al. [9] recommended movies based on above-average ratings by using collaborative filtering and the Pearson correlation coefficient. Last but not least, ChunYe Chien et al. [11] stressed efficient feature extraction and semantic understanding for better recommendations, especially in social media and movie-related situations, while Sushmita Roy et al. [10] addressed information overload using a variety of filtering techniques.

2.2 Importance of ML in Movie Recommendation

Machine Learning allows an organization on forecasting the interests of different users. Machine Learning provides multiple number of models to generalize it to the unseen data from the same population and measuring the performance. By implementing such models using machine learning in an application, the users get attracted to the particular application. Along with meeting the organization objective, the model should take care of accuracy, execution time, complexity and scalability as well to be considered as best model.

Machine learning (ML) stands as the cornerstone of modern movie recommendation systems, revolutionizing how users discover and engage with content. These systems harness the power of ML algorithms to analyze vast repositories of user data, including viewing history, ratings, and preferences, to deliver personalized movie recommendations tailored to individual tastes. By leveraging sophisticated techniques such as collaborative filtering and content- based filtering, ML algorithms can uncover intricate patterns and relationships within movie data, enabling accurate predictions of user preferences. This personalized approach not only enhances the user experience by offering relevant and engaging recommendations but also drives increased user engagement and satisfaction, ultimately leading to improved retention and revenue generation for streaming platforms and movie studios.

2.3 Implementation of machine learning using Python

Python is a popular programming language. It was created in 1991 by Guido van Rossum. It is used for:

1. web development (server-side),
2. software development,
3. mathematics,
4. system scripting.

The most recent major version of Python is Python 3. However, Python 2, although not being updated with anything other than security updates, is still quite popular. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse, Anaconda which are particularly useful when managing larger collections of Python files.

Python was designed for its readability. Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses. Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

In the older days, people used to perform Machine Learning tasks manually by coding all the algorithms and mathematical and statistical formula. This made the process time consuming, tedious and inefficient. But in the modern days, it is become very much easy and efficient compared to the olden days by various python libraries, frameworks, and modules. Today, Python is one of the most popular programming languages for this task and it has replaced many languages in the industry, one of the reason is its vast collection of libraries. Python libraries that used in Machine Learning are:

1. Numpy
2. Scikit-learn
3. Pandas
4. Flask

1. NumPy: is a very popular python library for large multi-dimensional array and matrix processing, with the help of a large collection of high-level mathematical functions. It is very useful for fundamental scientific computations in Machine Learning. It is particularly useful for linear algebra, Fourier transform, and random number capabilities. High-end libraries like TensorFlow uses NumPy internally for manipulation of Tensors.

2. Scikit-learn: is one of the most popular Machine Learning libraries for classical Machine Learning algorithms. It is built on top of two basic Python libraries, NumPy and SciPy. Scikit-learn supports most of the supervised and unsupervised learning algorithms. Scikit learn can also be used for data- mining and data-analysis, which makes it a great tool who is starting out with Machine Learning.

3. Pandas: is a popular Python library for data analysis. It is not directly related to Machine Learning. As we know that the dataset must be prepared before training. In this case, Pandas comes handy as it was developed specifically for data extraction and preparation. It provides high-level data structures and wide variety tools for data analysis. It provides many inbuilt methods for groping, combining and filtering data.

4. Flask: is a popular Python web framework for creating online applications, including ones that use Machine Learning models, is called Flask. Flask is a useful tool for deploying Machine Learning models as APIs (Application Programming Interfaces) since it offers a lightweight and easy-to-use method for creating web services. Flask is the name of the framework, which includes models for machine learning. Flask provides an easy-to-use and lightweight web service creation approach, making it a valuable tool for implementing Machine Learning models as APIs. The platform is known as Flask, and it includes models for machine learning. Flask provides a lightweight and user-friendly web service creation mechanism, making it a valuable tool for implementing Machine Learning models as APIs (Application Programming Interfaces).

2.4 Applications of Machine Learning

Streaming Platforms:

Streaming services like Netflix, Amazon Prime Video, and Hulu utilize machine learning algorithms to recommend movies and TV shows to users based on their viewing history, preferences, and behavior patterns.

Movie Rating Websites:

Websites like IMDb and Rotten Tomatoes employ machine learning to suggest movies to users based on their ratings, reviews, and browsing history.

Mobile Apps:

Mobile apps dedicated to movie recommendations, such as Letterboxd and TMDb, leverage machine learning to provide personalized movie suggestions to users on-the-go.

Smart TVs:

Smart TV platforms incorporate machine learning algorithms to recommend movies and TV shows to users based on their viewing habits, preferences, and interactions with the TV interface.

Social Media Platforms:

Social media platforms like Facebook and Twitter utilize machine learning to recommend movies to users based on their social connections, interests, and engagement with movie-related content.

E-commerce Platforms:

E-commerce platforms like Amazon and eBay use machine learning to recommend movies to users based on their purchase history, browsing behavior, and preferences.

Voice Assistants:

Voice-activated assistants like Amazon Alexa and Google Assistant employ machine learning algorithms to provide personalized movie recommendations to users based on their voice commands, preferences, and context.

Video Game Consoles:

Video game consoles like PlayStation and Xbox incorporate machine learning to recommend movies and TV shows to users based on their gaming preferences, activity history, and gaming behavior.

Cinema Chains:

Cinema chains and movie theaters use machine learning to recommend movies to customers based on their ticket purchases, movie preferences, and demographic information.

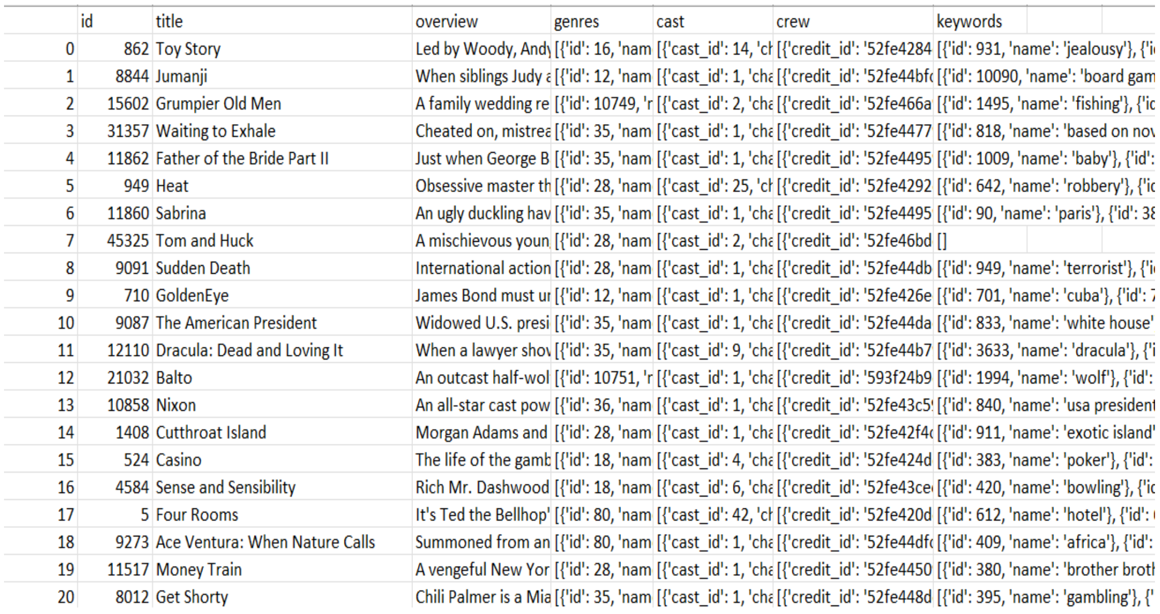
3. SYSTEM ANALYSIS & DESIGN

3.1 Scope of the project

The scope of the Movie Recommendation System project encompasses the development of a comprehensive recommendation engine capable of providing personalized movie suggestions to users based on their preferences and past interactions. This includes the implementation of various recommendation algorithms such as content-based, collaborative filtering, and hybrid approaches to cater to different user scenarios and preferences. Additionally, the project involves the creation of a user-friendly frontend interface using the Flask web framework, allowing users to easily interact with the system and receive tailored movie recommendations. The documentation will cover the architecture, design, implementation details, deployment procedures, and testing strategies involved in building and deploying the Movie Recommendation System, providing a comprehensive understanding of its scope and functionality.

3.2 Analysis

The Fig 3.1 is the dataset contains 7 attributes which are used to predict the movie recommendation system such as



	id	title	overview	genres	cast	crew	keywords
0	862	Toy Story	Led by Woody, Andy	[{'id': 16, 'nam	[{'cast_id': 14, 'ch	[{'credit_id': '52fe4284	[{'id': 931, 'name': 'jealousy'}, {'i
1	8844	Jumanji	When siblings Judy a	[{'id': 12, 'nam	[{'cast_id': 1, 'cha	[{'credit_id': '52fe44bf	[{'id': 10090, 'name': 'board gam
2	15602	Grumpier Old Men	A family wedding re	[{'id': 10749, 'r	[{'cast_id': 2, 'cha	[{'credit_id': '52fe466a	[{'id': 1495, 'name': 'fishing'}, {'ic
3	31357	Waiting to Exhale	Cheated on, mistrea	[{'id': 35, 'nam	[{'cast_id': 1, 'cha	[{'credit_id': '52fe4477	[{'id': 818, 'name': 'based on nov
4	11862	Father of the Bride Part II	Just when George B	[{'id': 35, 'nam	[{'cast_id': 1, 'cha	[{'credit_id': '52fe4495	[{'id': 1009, 'name': 'baby'}, {'id':
5	949	Heat	Obsessive master th	[{'id': 28, 'nam	[{'cast_id': 25, 'ch	[{'credit_id': '52fe4292	[{'id': 642, 'name': 'robbery'}, {'ic
6	11860	Sabrina	An ugly duckling hav	[{'id': 35, 'nam	[{'cast_id': 1, 'cha	[{'credit_id': '52fe4495	[{'id': 90, 'name': 'paris'}, {'id': 38
7	45325	Tom and Huck	A mischievous youn	[{'id': 28, 'nam	[{'cast_id': 2, 'cha	[{'credit_id': '52fe46bd	[]
8	9091	Sudden Death	International action	[{'id': 28, 'nam	[{'cast_id': 1, 'cha	[{'credit_id': '52fe44db	[{'id': 949, 'name': 'terrorist'}, {'i
9	710	GoldenEye	James Bond must ur	[{'id': 12, 'nam	[{'cast_id': 1, 'cha	[{'credit_id': '52fe426e	[{'id': 701, 'name': 'cuba'}, {'id': 7
10	9087	The American President	Widowed U.S. presi	[{'id': 35, 'nam	[{'cast_id': 1, 'cha	[{'credit_id': '52fe44da	[{'id': 833, 'name': 'white house'
11	12110	Dracula: Dead and Loving It	When a lawyer shov	[{'id': 35, 'nam	[{'cast_id': 9, 'cha	[{'credit_id': '52fe44b7	[{'id': 3633, 'name': 'dracula'}, {'i
12	21032	Balto	An outcast half-wol	[{'id': 10751, 'r	[{'cast_id': 1, 'cha	[{'credit_id': '593f24b9	[{'id': 1994, 'name': 'wolf'}, {'id':
13	10858	Nixon	An all-star cast pow	[{'id': 36, 'nam	[{'cast_id': 1, 'cha	[{'credit_id': '52fe43c5	[{'id': 840, 'name': 'usa president
14	1408	Cutthroat Island	Morgan Adams and	[{'id': 28, 'nam	[{'cast_id': 1, 'cha	[{'credit_id': '52fe42f4	[{'id': 911, 'name': 'exotic island'
15	524	Casino	The life of the gamb	[{'id': 18, 'nam	[{'cast_id': 4, 'cha	[{'credit_id': '52fe424d	[{'id': 383, 'name': 'poker'}, {'id':
16	4584	Sense and Sensibility	Rich Mr. Dashwood	[{'id': 18, 'nam	[{'cast_id': 6, 'cha	[{'credit_id': '52fe43ce	[{'id': 420, 'name': 'bowling'}, {'ic
17	5	Four Rooms	It's Ted the Bellhop'	[{'id': 80, 'nam	[{'cast_id': 42, 'ch	[{'credit_id': '52fe420d	[{'id': 612, 'name': 'hotel'}, {'id':
18	9273	Ace Ventura: When Nature Calls	Summoned from an	[{'id': 80, 'nam	[{'cast_id': 1, 'cha	[{'credit_id': '52fe44df	[{'id': 409, 'name': 'africa'}, {'id':
19	11517	Money Train	A vengeful New Yor	[{'id': 28, 'nam	[{'cast_id': 1, 'cha	[{'credit_id': '52fe4450	[{'id': 380, 'name': 'brother broth
20	8012	Get Shorty	Chili Palmer is a Mia	[{'id': 35, 'nam	[{'cast_id': 1, 'cha	[{'credit_id': '52fe448d	[{'id': 395, 'name': 'gambling'}, {'

Fig 3.1: Dataset before pre-processing

Id: Which is unique to each and every movie.

Title: Which denotes the title of the movie.

3.3 Data Preprocessing

Before feeding data to an algorithm, we have to apply transformations to our data which is referred as pre-processing. By performing pre-processing, the raw data which is not feasible for analysis is converted into clean data.

In-order to achieve better results using a model in Machine Learning, data format has to be in a proper manner. The data should be in a particular format for different algorithms. For example, if we consider Random Forest algorithm it does not support null values. So that those null values have to be managed using raw data.

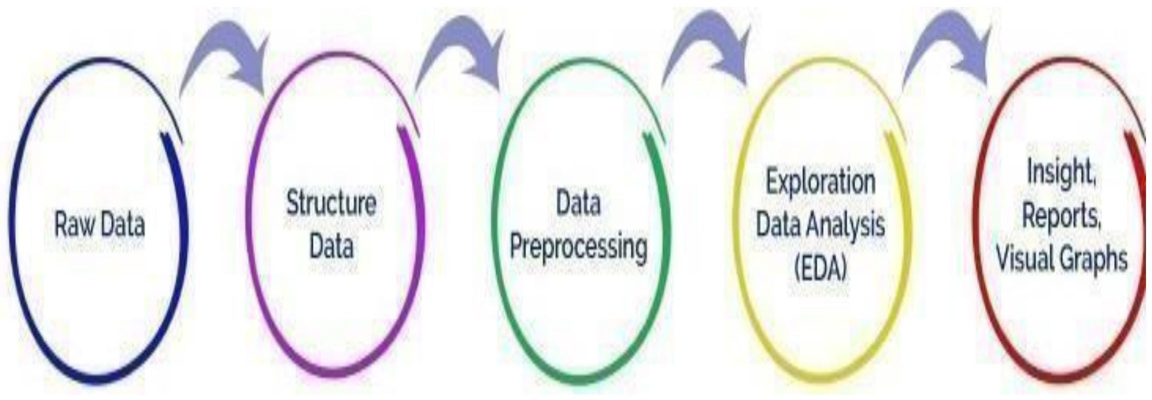

Raw Data
Structure Data
Data Preprocessing
Exploration Data Analysis (EDA)
Insight, Reports, Visual Graphs

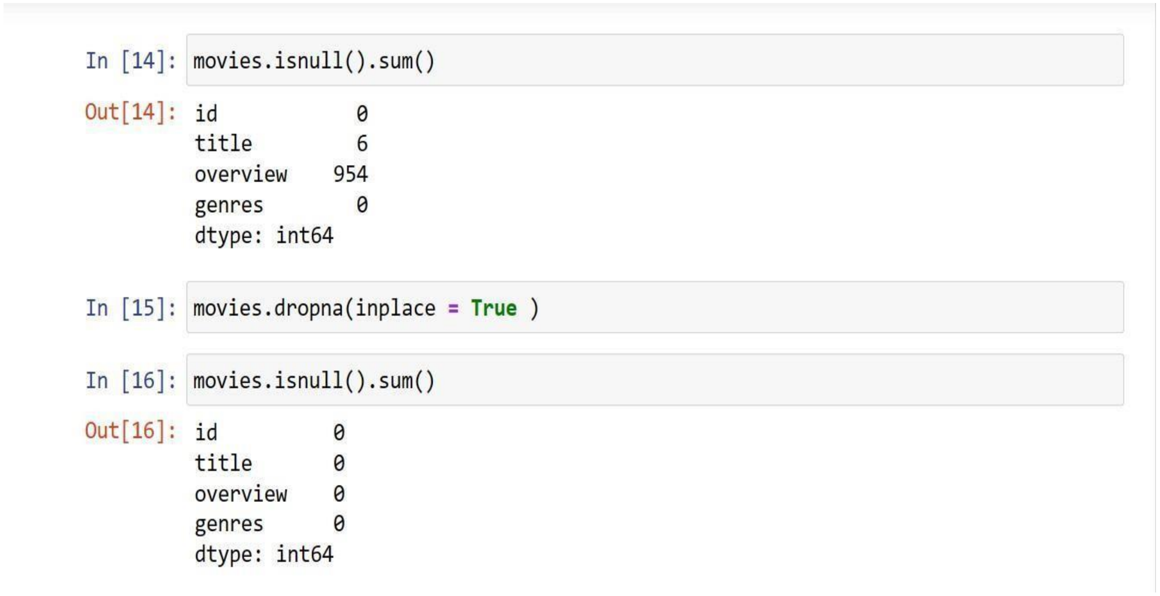
Fig 3.2: Data Pre-processing

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Pre-processing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. Fig 3.2 shows the flow of data pre-processing.

3.4 Need of Data Preprocessing

For achieving better results from the applied model in Machine Learning projects the format of the data has to be in a proper manner. Some specified Machine Learning model needs information in a specified format. For example, Random Forest algorithm does not support null values, therefore to execute random forest algorithm null values have to be managed from the original raw data set. Another aspect is that data set should be formatted in such a way that more than one Machine Learning and Deep Learning algorithms are executed in one data set, and best out of them is chosen.

3.5 Missing values

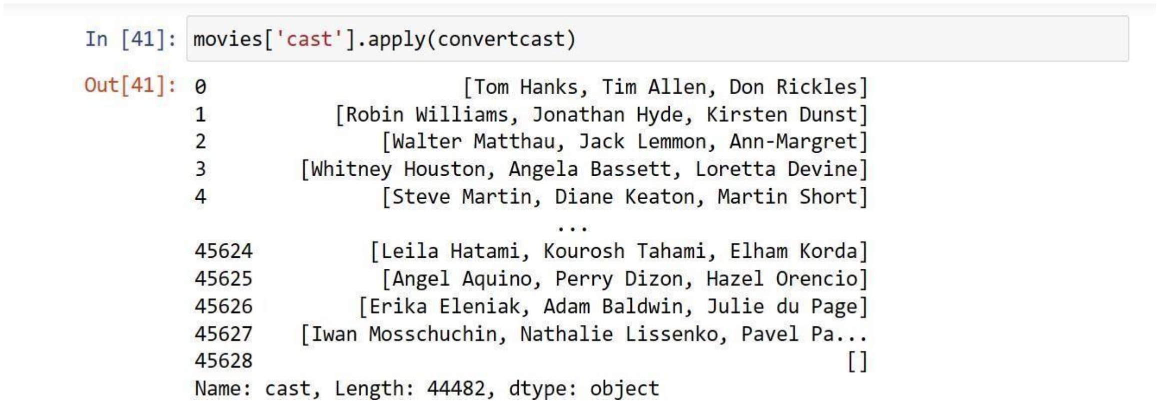


```
In [14]: movies.isnull().sum()

Out[14]: id            0
         title         6
         overview    954
         genres        0
         dtype: int64

In [15]: movies.dropna(inplace = True )

In [16]: movies.isnull().sum()

Out[16]: id          0
         title       0
         overview    0
         genres      0
         dtype: int64
```

Fig 3.3: Screenshot of removing null values

Filling missing values is one of the pre-processing techniques. The missing values in the dataset is represented as '?' but it a non-standard missing value and it has to be converted into a standard missing value NaN. So that pandas can detect the missing values. We are dropping those missing values. The Fig 3.3 shows the dropping of null values in the taken dataset.

3.6 Overview of datasets

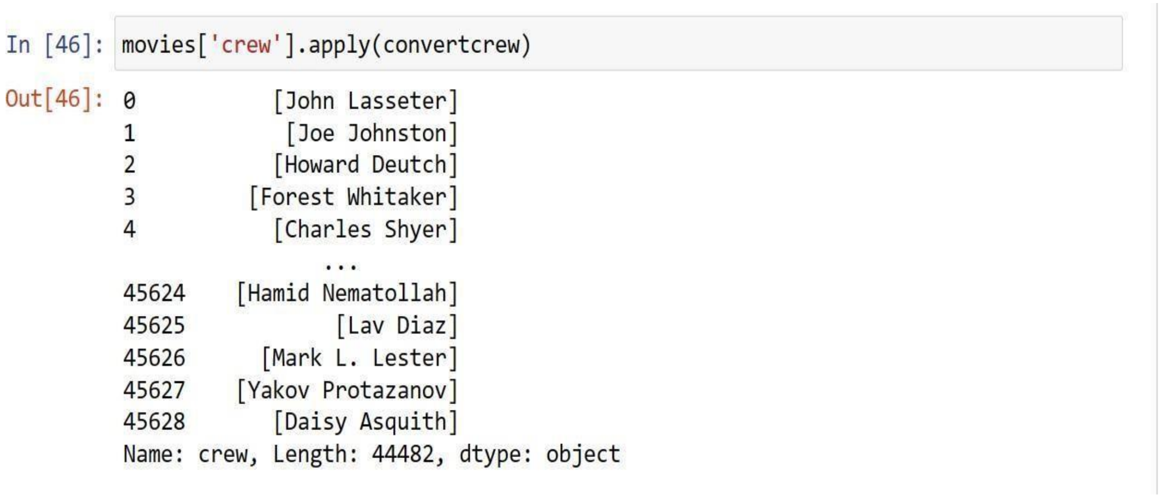


```
In [41]: movies['cast'].apply(convertcast)

Out[41]: 0                        [Tom Hanks, Tim Allen, Don Rickles]
         1              [Robin Williams, Jonathan Hyde, Kirsten Dunst]
         2                  [Walter Matthau, Jack Lemmon, Ann-Margret]
         3           [Whitney Houston, Angela Bassett, Loretta Devine]
         4                  [Steve Martin, Diane Keaton, Martin Short]
                                        ...
         45624             [Leila Hatami, Kourosh Tahami, Elham Korda]
         45625               [Angel Aquino, Perry Dizon, Hazel Orencio]
         45626             [Erika Eleniak, Adam Baldwin, Julie du Page]
         45627       [Iwan Mosschuchin, Nathalie Lissenko, Pavel Pa...
         45628                                                      []
         Name: cast, Length: 44482, dtype: object
```

Fig 3.4: Screenshot of cast attribute after preprocessing

Cast: Cast attribute consists the actors and actresses who appear in the movie. Fig 3.4 gives you the type of information and structure of information present in cast attribute after pre-processing.

Crew:

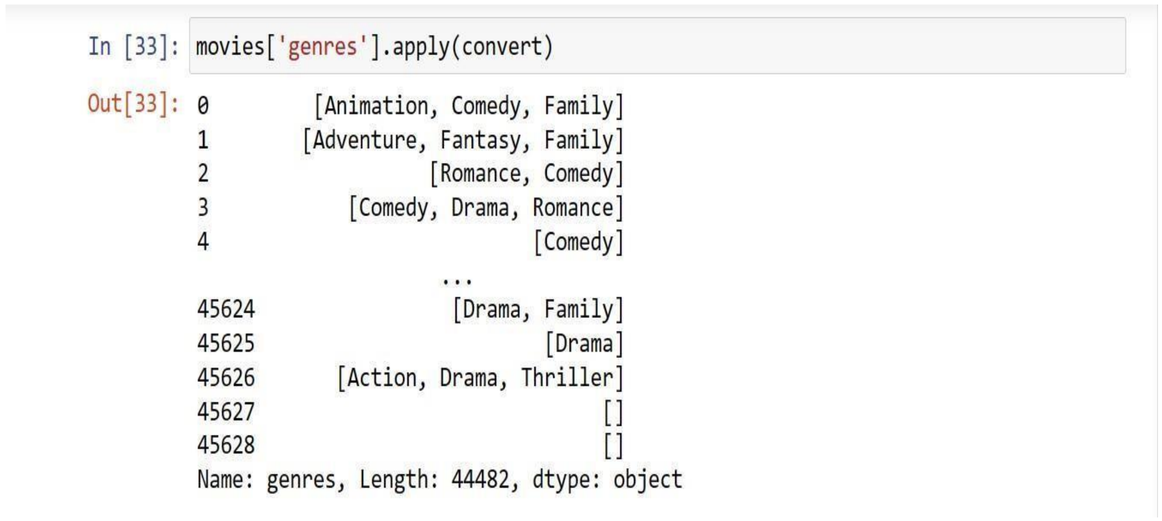


```
In [46]: movies['crew'].apply(convertcrew)

Out[46]: 0             [John Lasseter]
         1              [Joe Johnston]
         2             [Howard Deutch]
         3           [Forest Whitaker]
         4             [Charles Shyer]
                            ...
         45624    [Hamid Nematollah]
         45625            [Lav Diaz]
         45626       [Mark L. Lester]
         45627    [Yakov Protazanov]
         45628       [Daisy Asquith]
         Name: crew, Length: 44482, dtype: object
```

Fig 3.5: Screenshot of crew attribute after pre-processing

The Crew attribute includes the directors, producers, writers and people behind the movie. Fig 3.5 gives you the type of information and structure of information present in crew attribute after pre- processing.

Genres:

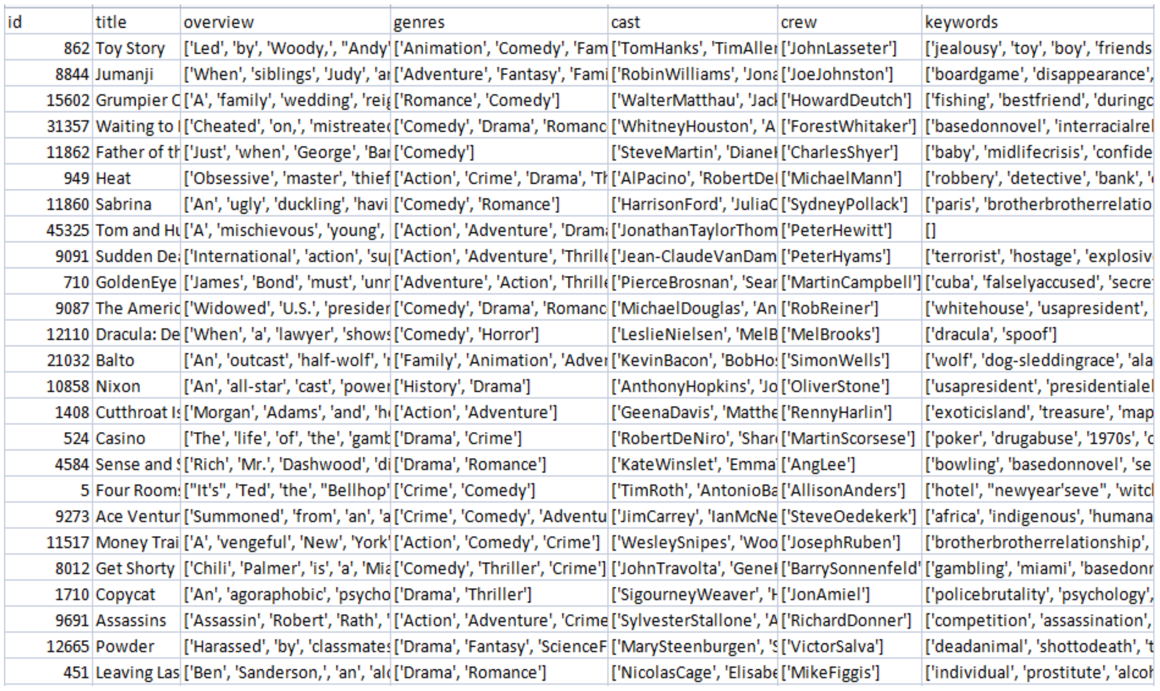


```
In [33]: movies['genres'].apply(convert)

Out[33]: 0         [Animation, Comedy, Family]
         1        [Adventure, Fantasy, Family]
         2                   [Romance, Comedy]
         3             [Comedy, Drama, Romance]
         4                            [Comedy]
                             ...
         45624                 [Drama, Family]
         45625                         [Drama]
         45626       [Action, Drama, Thriller]
         45627                              []
         45628                              []
         Name: genres, Length: 44482, dtype: object
```

Fig 3.6: Screenshot of genres attribute after pre-processing

Genres attribute describes the genre of the movie. Fig 3.6 gives you the type of information and structure of information present in genres attribute after pre-processing.

Overview:

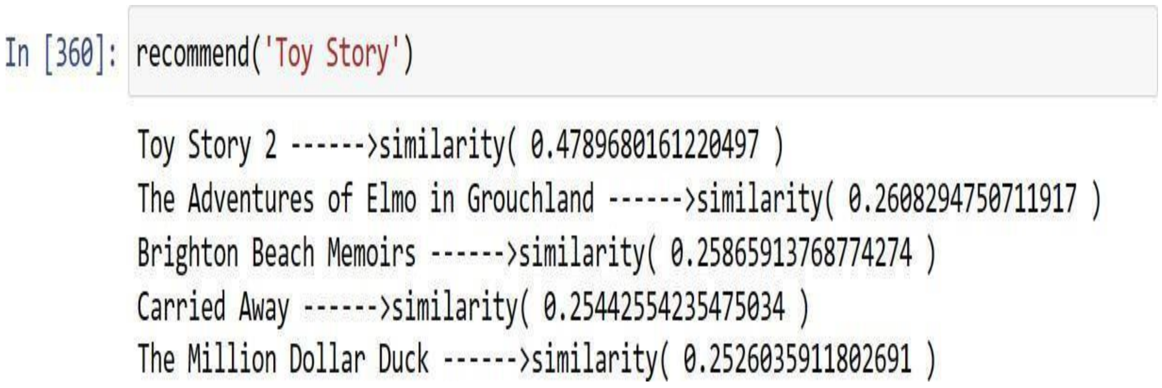


id	title	overview	genres	cast	crew	keywords
862	Toy Story	['Led', 'by', 'Woody,', "Andy'	['Animation', 'Comedy', 'Fam	['TomHanks', 'TimAller	['JohnLasseter']	['jealousy', 'toy', 'boy', 'friends
8844	Jumanji	['When', 'siblings', 'Judy', 'ar	['Adventure', 'Fantasy', 'Fami	['RobinWilliams', 'Jona	['JoeJohnston']	['boardgame', 'disappearance',
15602	Grumpier C	['A', 'family', 'wedding', 'reig	['Romance', 'Comedy']	['WalterMatthau', 'Jacl	['HowardDeutch']	['fishing', 'bestfriend', 'duringc
31357	Waiting to I	['Cheated', 'on,', 'mistreatec	['Comedy', 'Drama', 'Romanc	['WhitneyHouston', 'A	['ForestWhitaker']	['basedonnovel', 'interracialrel
11862	Father of th	['Just', 'when', 'George', 'Bai	['Comedy']	['SteveMartin', 'DianeI	['CharlesShyer']	['baby', 'midlifecrisis', 'confide
949	Heat	['Obsessive', 'master', 'thief	['Action', 'Crime', 'Drama', 'Tl	['AlPacino', 'RobertDeI	['MichaelMann']	['robbery', 'detective', 'bank', '
11860	Sabrina	['An', 'ugly', 'duckling', 'havi	['Comedy', 'Romance']	['HarrisonFord', 'JuliaC	['SydneyPollack']	['paris', 'brotherbrotherrelatio
45325	Tom and Hu	['A', 'mischievous', 'young',	['Action', 'Adventure', 'Drama	['JonathanTaylorThom	['PeterHewitt']	[]
9091	Sudden Dea	['International', 'action', 'su	['Action', 'Adventure', 'Thrille	['Jean-ClaudeVanDam	['PeterHyams']	['terrorist', 'hostage', 'explosiv
710	GoldenEye	['James', 'Bond', 'must', 'unr	['Adventure', 'Action', 'Thrille	['PierceBrosnan', 'Sear	['MartinCampbell']	['cuba', 'falselyaccused', 'secre
9087	The Americ	['Widowed', 'U.S.', 'presider	['Comedy', 'Drama', 'Romanc	['MichaelDouglas', 'An	['RobReiner']	['whitehouse', 'usapresident',
12110	Dracula: De	['When', 'a', 'lawyer', 'shows	['Comedy', 'Horror']	['LeslieNielsen', 'MelB	['MelBrooks']	['dracula', 'spoof']
21032	Balto	['An', 'outcast', 'half-wolf', 'r	['Family', 'Animation', 'Adver	['KevinBacon', 'BobHo:	['SimonWells']	['wolf', 'dog-sleddingrace', 'ala
10858	Nixon	['An', 'all-star', 'cast', 'power	['History', 'Drama']	['AnthonyHopkins', 'Jo	['OliverStone']	['usapresident', 'presidentialel
1408	Cutthroat Is	['Morgan', 'Adams', 'and', 'h	['Action', 'Adventure']	['GeenaDavis', 'Matthe	['RennyHarlin']	['exoticisland', 'treasure', 'map
524	Casino	['The', 'life', 'of', 'the', 'gamb	['Drama', 'Crime']	['RobertDeNiro', 'Shar	['MartinScorsese']	['poker', 'drugabuse', '1970s', 'c
4584	Sense and S	['Rich', 'Mr.', 'Dashwood', 'di	['Drama', 'Romance']	['KateWinslet', 'Emma	['AngLee']	['bowling', 'basedonnovel', 'se
5	Four Room:	["It's", 'Ted', 'the', "Bellhop'	['Crime', 'Comedy']	['TimRoth', 'AntonioBa	['AllisonAnders']	['hotel', "newyear'seve", 'witcl
9273	Ace Ventur	['Summoned', 'from', 'an', 'a	['Crime', 'Comedy', 'Adventu	['JimCarrey', 'IanMcNe	['SteveOedekerk']	['africa', 'indigenous', 'humana
11517	Money Trai	['A', 'vengeful', 'New', 'York'	['Action', 'Comedy', 'Crime']	['WesleySnipes', 'Woo	['JosephRuben']	['brotherbrotherrelationship',
8012	Get Shorty	['Chili', 'Palmer', 'is', 'a', 'Mia	['Comedy', 'Thriller', 'Crime']	['JohnTravolta', 'GeneI	['BarrySonnenfeld'	['gambling', 'miami', 'basedonr
1710	Copycat	['An', 'agoraphobic', 'psycho	['Drama', 'Thriller']	['SigourneyWeaver', 'H	['JonAmiel']	['policebrutality', 'psychology',
9691	Assassins	['Assassin', 'Robert', 'Rath', '	['Action', 'Adventure', 'Crime	['SylvesterStallone', 'A	['RichardDonner']	['competition', 'assassination',
12665	Powder	['Harassed', 'by', 'classmates	['Drama', 'Fantasy', 'ScienceF	['MarySteenburgen', 'S	['VictorSalva']	['deadanimal', 'shottodeath', 't
451	Leaving Las	['Ben', 'Sanderson,', 'an', 'alc	['Drama', 'Romance']	['NicolasCage', 'Elisabe	['MikeFiggis']	['individual', 'prostitute', 'alcoh

Fig:3.7 Dataset after pre-processing

Overview describes the brief summery about the movie. Fig 3.7 gives you the type of information and structure of information present in genres attribute after pre-processing. After completion of data pre-processing the data set looks like of Fig 3.7. Consider Fig 3.7 for better understanding.

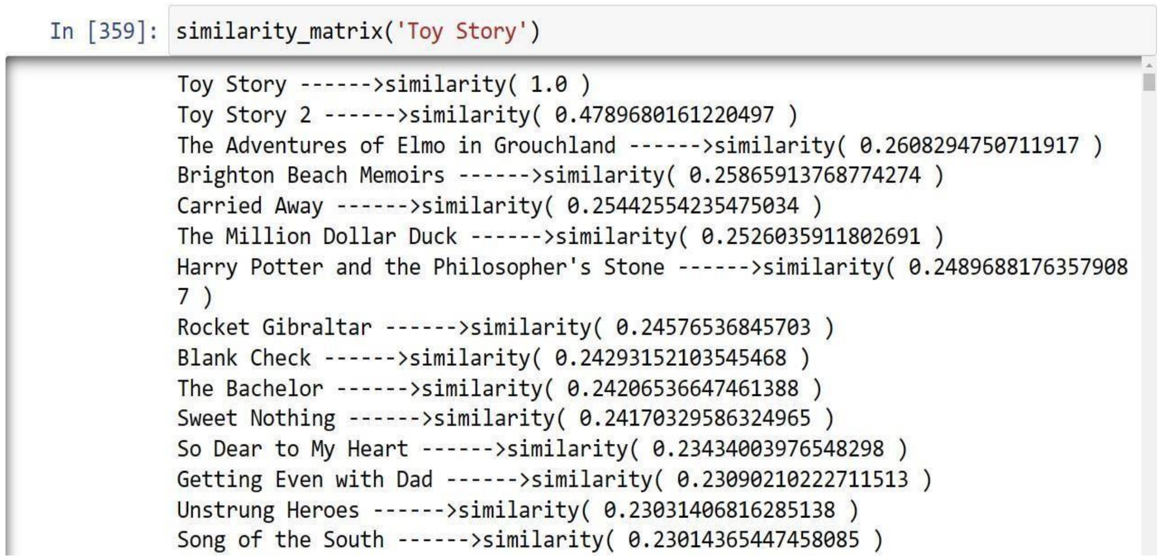
3.7 Cross Validation:



```
In [360]: recommend('Toy Story')

Toy Story 2 ------>similarity( 0.4789680161220497 )
The Adventures of Elmo in Grouchland ------>similarity( 0.2608294750711917 )
Brighton Beach Memoirs ------>similarity( 0.25865913768774274 )
Carried Away ------>similarity( 0.25442554235475034 )
The Million Dollar Duck ------>similarity( 0.2526035911802691 )
```

Fig 3.8: Movies recommended by implemented model

Cross-validation is a technique in which we train our model using the subset of the dataset and then evaluate using the complementary subset of the data-set. The two steps involved in cross-validation are as follows:



```
In [359]: similarity_matrix('Toy Story')

Toy Story ------>similarity( 1.0 )
Toy Story 2 ------>similarity( 0.4789680161220497 )
The Adventures of Elmo in Grouchland ------>similarity( 0.2608294750711917 )
Brighton Beach Memoirs ------>similarity( 0.25865913768774274 )
Carried Away ------>similarity( 0.25442554235475034 )
The Million Dollar Duck ------>similarity( 0.2526035911802691 )
Harry Potter and the Philosopher's Stone ------>similarity( 0.2489688176357908
7 )
Rocket Gibraltar ------>similarity( 0.24576536845703 )
Blank Check ------>similarity( 0.24293152103545468 )
The Bachelor ------>similarity( 0.24206536647461388 )
Sweet Nothing ------>similarity( 0.24170329586324965 )
So Dear to My Heart ------>similarity( 0.23434003976548298 )
Getting Even with Dad ------>similarity( 0.23090210222711513 )
Unstrung Heroes ------>similarity( 0.23031406816285138 )
Song of the South ------>similarity( 0.23014365447458085 )
```

Fig 3.9: Printing the similarity for every movie in the dataset

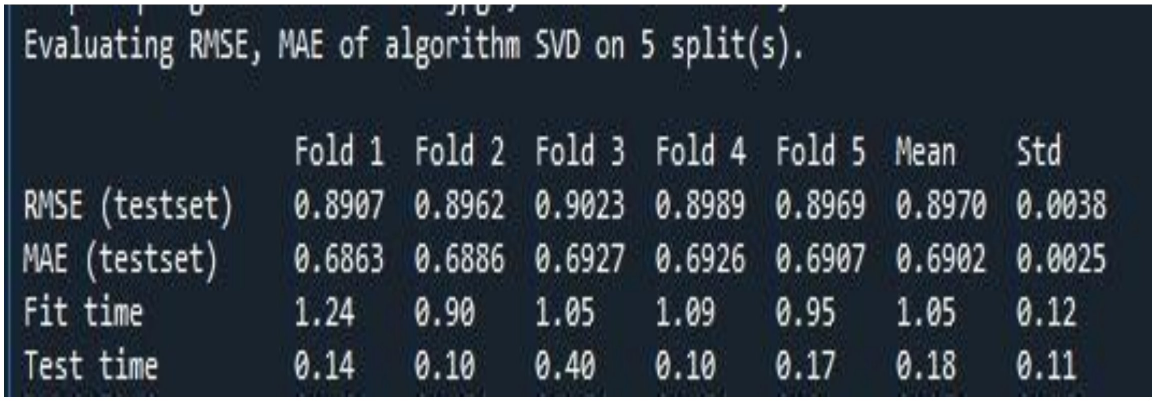
- Recommending movies from the model.
- Displaying the similarity to each and every movie present in dataset.
- Comparing the recommended movies with similarity of the movies.

Fig 3.8 is the screenshot of recommendations of a selected movie or given movie and it is cross checked by displaying the similarity. The Fig 3.9 shows the similarity between the movies.

3.8 RMSE Scores of SVD model

We compute the SVD approach's root mean square error (RMSE) and mean absolute error (MAE), yielding RMSE scores of 0.89 and 0.68, respectively.

We thoroughly calculate two important prediction accuracy measures, the Root Mean Square Error (RMSE) and the Mean Absolute Error (MAE), in order to assess the effectiveness of our Singular Value Decomposition (SVD) technique in the collaborative filtering part of our hybrid movie recommendation system. The acquired RMSE score of 0.89 and MAE score of 0.68 serve as important reference points for evaluating how well the system predicts user movie evaluations.



```
Evaluating RMSE, MAE of algorithm SVD on 5 split(s).

                  Fold 1  Fold 2  Fold 3  Fold 4  Fold 5  Mean    Std
RMSE (testset)    0.8907  0.8962  0.9023  0.8989  0.8969  0.8970  0.0038
MAE (testset)     0.6863  0.6886  0.6927  0.6926  0.6907  0.6902  0.0025
Fit time          1.24    0.90    1.05    1.09    0.95    1.05    0.12
Test time         0.14    0.10    0.40    0.10    0.17    0.18    0.11
```

Fig 3.10: RMSE scores of SVD model

The RMSE measure provides information about the correctness of the model by punishing greater mistakes more harshly. It does this by taking into account the square root of the average squared discrepancies between projected and actual ratings. With an RMSE of 0.89, it can be inferred that the average deviation between the actual user ratings and the predictions made by the SVD model is less than one rating point. This degree of precision demonstrates how well the model can capture the subtleties of user preferences and the underlying trends in movie ratings.

Similarly, without squaring the ratings, the MAE score which determines the average absolute difference between expected and observed ratings offers a simple way to gauge prediction mistakes. The model's ability to provide predictions that closely match users' actual evaluations is indicated by the resultant score of 0.68, which emphasizes the model's dependability in a more understandable way than RMSE.

When taken as a whole, these measures show off the SVD algorithm's predicted accuracy as well as its usefulness in real-world recommendation systems. The SVD's capacity to reduce rating prediction mistakes and improve user experience with individualized recommendations is demonstrated by the comparatively low values of RMSE and MAE. Through the use of latent factor analysis, the SVD- based collaborative filtering method successfully deciphers the intricate web of user-item interactions to guarantee that recommendations are both relevant and customized to each user's profile, resulting in a more enjoyable and fulfilling movie discovery process.

3.9 Weighted Rating

The weighted rating formula is a method used to determine the overall rating or score of an item, taking into account both the average rating and the number of ratings it has received. It is commonly used in recommendation systems, particularly in scenarios where items have varying numbers of ratings.

The formula incorporates two key components:

1. Average Rating (R): This represents the average rating or score of the item, calculated by summing up all individual ratings and dividing by the total number of ratings.
2. Weighting Factor (v / (v + m)): This factor accounts for the number of ratings the item has received relative to a predetermined threshold (m). The purpose of this factor is to give more weight to items with a higher number of ratings, indicating higher confidence in the average rating.

The weighted rating formula is typically expressed as follows:

$$\textbf{Weighted Rating (WR)} = (v / (v + m)) \cdot R + (m / (v + m)) \cdot C$$

Where:

- WR is the weighted rating of the item.
- v is the total number of ratings the item has received.
- m is the minimum number of ratings required for the item to be considered in the calculation.
- R is the average rating of the item.
- C is the average rating across all items.

In essence, the weighted rating formula ensures that items with a higher number of ratings are given more weight in the overall rating calculation, while also accounting for the average rating of the item and the average rating across all items. This helps mitigate biases that may arise from variations in the number of ratings received by different items, ensuring a more fair and reliable assessment of item quality.

4.IMPLEMENTATION

4.1 Movie Recommender.ipynb

# Importing Libraries

```
%matplotlib inline
import matplotlib.pyplot as plt import seaborn as sns
import pandas as pd import numpy as np import ast
from scipy import stats from ast import literal_eval
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from sklearn.metrics.pairwise import linear_kernel, cosine_similarity
from nltk.stem.snowball import SnowballStemmer
from nltk.stem.wordnet import WordNetLemmatizer
from nltk.corpus import wordnet
from surprise import Reader, Dataset, SVD
from surprise.model_selection import cross_validate
import warnings;
warnings.simplefilter('ignore')
conda install -c conda-forge scikit-surprise
conda install -c conda-forge textblob
```

# Loading Datasets

```
credits = pd.read_csv('./archive/credits.csv')
keywords = pd.read_csv('./archive/keywords.csv')
links_small = pd.read_csv('./archive/links_small.csv')
md = pd.read_csv('./archive/movies_metadata.csv',low_memory=False)
ratings = pd.read_csv('./archive/ratings_small.csv')
```

# Exploratory Data Analysis

```
credits.head()
credits.head()
credits.shape
credits.info()
```

```
keywords.head()
keywords.columns
keywords.shape
keywords.info()
links_small.head()
links_small.columns
links_small.shape
links_small.info()
md.iloc[0:3].transpose()
md.columns
md.shape
md.info()
ratings.head()
ratings.columns
ratings.shape
ratings.info()
md['genres'] = md['genres'].fillna('[]').apply(literal_eval).apply(lambda x: [i[ 'name'] for i
in x] if isinstance(x, list) else [])
```

# this is V

```
vote_counts = md[md['vote_count'].notnull()]['vote_count'].astype('int')
```

# this is R

```
vote_averages = md[md['vote_average'].notnull()]['vote_average'].astype('int')
```

# this is C

```
C = vote_averages.mean()
```

# Pre-processing step for getting year from date by splliting it using '-'

```
md['year'] = pd.to_datetime(md['release_date'], errors='coerce').apply( lambda x:
    str(x).split('-')[0] if x != np.nan else np.nan)
qualified = md[(md['vote_count'] >= m) & (md['vote_count'].notnull()) &
            (md['vote_average'].notnull())][['title','year', 'vote_count', 'vote_average',
            'popularity', 'genres']]
```

```
qualified['vote_count'] = qualified['vote_count'].astype('int')
qualified['vote_average'] = qualified['vote_average'].astype('int')
qualified.shape
def weighted_rating(x):
    v = x['vote_count']
    R = x['vote_average']

     return (v/(v+m) * R) + (m/(m+v) * C)

qualified['wr'] = qualified.apply(weighted_rating, axis=1)

qualified = qualified.sort_values('wr', ascending=False).head(250)

qualified.head(15)
s = md.apply(lambda x: pd.Series(x['genres']),axis=1).stack().reset_index(level=1,
drop=True)
s.name = 'genre'
gen_md = md.drop('genres', axis=1).join(s)
gen_md.head(3).transpose()
def build_chart(genre, percentile=0.85):
   df = gen_md[gen_md['genre'] == genre]
  vote_counts = df[df['vote_count'].notnull()]['vote_count'].astype('int')
  vote_averages = df[df['vote_average'].notnull()]['vote_average'].astype('int')
  C = vote_averages.mean()
  m = vote_counts.quantile(percentile)
  qualified = df[(df['vote_count'] >= m) & (df['vote_count'].notnull()) &
                (df['vote_average'].notnull())][['title', 'year', 'vote_count', 'vote_average',
                'popularity']]
  qualified['vote_count'] = qualified['vote_count'].astype('int')
  qualified['vote_average'] = qualified['vote_average'].astype('int')
  qualified['wr'] = qualified.apply(lambda x:
  (x['vote_count']/(x['vote_count']+m) * x['vote_average']) + (m/(m+x['vote_count']) *C),
  axis=1)
  qualified = qualified.sort_values('wr', ascending=False).head(250)
  return qualified build_chart('Romance').head(15)
```

# Content based recommendation system

```
links_small = links_small[links_small['tmdbId'].notnull()]['tmdbId'].astype('int')
```

# Pre-processing

```
def convert_int(x):
 try:
    return int(x)
 except:
     return np.nan
md['id'] = md['id'].apply(convert_int)
md[md['id'].isnull()]
md = md.drop([19730, 29503, 35587])
md['id'] = md['id'].astype('int')
smd = md[md['id'].isin(links_small)] smd.shape
```

# Content based recommendation system : Using movie description and taglines

```
smd['tagline'] = smd['tagline'].fillna('')
smd['description'] = smd['overview'] + smd['tagline']
smd['description'] = smd['description'].fillna('')
tf = TfidfVectorizer(analyzer='word',ngram_range=(1, 2),min_df=0, stop_words='english')
tfidf_matrix = tf.fit_transform(smd['description'])
tfidf_matrix.shape
cosine_sim = linear_kernel(tfidf_matrix, tfidf_matrix)
cosine_sim[0]
 #cosine_sim.shape
smd = smd.reset_index()
titles = smd['title']
```

```
indices = pd.Series(smd.index, index=smd['title'])
 #indices.head(2)
def get_recommendations(title):
    idx = indices[title]
    sim_scores = list(enumerate(cosine_sim[idx]))
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
     sim_scores = sim_scores[1:31]
    movie_indices = [i[0] for i in sim_scores]
    return titles.iloc[movie_indices]
get_recommendations('The Godfather').head(10)
```

# Content based RS : Using movie description, taglines, cast, director and genres

```
keywords['id'] = keywords['id'].astype('int')
credits['id'] = credits['id'].astype('int')
md['id'] = md['id'].astype('int')
md.shape
md = md.merge(credits, on='id')
md = md.merge(keywords, on='id')
smd = md[md['id'].isin(links_small)]
smd.shape
smd['cast'] = smd['cast'].apply(literal_eval)
smd['crew'] = smd['crew'].apply(literal_eval)
smd['keywords'] = smd['keywords'].apply(literal_eval)
smd['cast_size'] = smd['cast'].apply(lambda x: len(x))
smd['crew_size'] = smd['crew'].apply(lambda x: len(x))
def get_director(x):
    for i in x:
        if i['job'] == 'Director':
            return i['name']
    return np.nan
```

```
smd['director'] = smd['crew'].apply(get_director)
smd['cast'] = smd['cast'].apply(lambda x: [i['name'] for i in x] if isinstance(x, list) else [])
smd['cast'] = smd['cast'].apply(lambda x: x[:3] if len(x) >=3 else x)
smd['keywords'] = smd['keywords'].apply(lambda x: [i['name'] for i in x] if isinstance(x,
list) else [])
smd['cast'] = smd['cast'].apply(lambda x: [str.lower(i.replace(" ", "")) for i in x])
smd['director'] = smd['director'].astype('str').apply(lambda x: str.lower(x.replace(" ", "")))
smd['director'] = smd['director'].apply(lambda x: [x,x, x])
s = smd.apply(lambda x: pd.Series(x['keywords']),axis=1).stack().reset_index(level=1,
drop=True)
s.name = 'keyword'
s = s.value_counts()
s[:5]
s = s[s > 1]
# Just an example
stemmer = SnowballStemmer('english')
stemmer.stem('dogs')
def filter_keywords(x):
    words = []
    for i in x:
        if i in s:
            words.append(i)
    return words
smd['keywords'] = smd['keywords'].apply(filter_keywords)
smd['keywords'] = smd['keywords'].apply(lambda x: [stemmer.stem(i) for i in x])
smd['keywords'] = smd['keywords'].apply(lambda x: [str.lower(i.replace(" ", "")) for i in
x])
smd['soup'] = smd['keywords'] + smd['cast'] + smd['director'] + smd['genres']
smd['soup'] = smd['soup'].apply(lambda x: ' '.join(x))
```

# CountVectorizer

```
count = CountVectorizer(analyzer='word',ngram_range=(1, 2),min_df=0,
stop_words='english')
count_matrix = count.fit_transform(smd['soup'])
cosine_sim = cosine_similarity(count_matrix, count_matrix)
smd = smd.reset_index()
titles = smd['title']
indices = pd.Series(smd.index, index=smd['title'])
get_recommendations('The  Godfather').head(10)
def improved_recommendations(title):
    idx = indices[title]
    sim_scores = list(enumerate(cosine_sim[idx]))
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
    sim_scores = sim_scores[1:26]
    movie_indices = [i[0] for i in sim_scores]
    movies = smd.iloc[movie_indices][['title', 'vote_count', 'vote_average', 'year']]
    vote_counts = movies[movies['vote_count'].notnull()]['vote_count'].astype('int')
    vote_averages = movies[movies['vote_average'].notnull()]['vote_average'].astype('int')
    C = vote_averages.mean()
    m = vote_counts.quantile(0.60)
    qualified = movies[(movies['vote_count'] >= m) & (movies['vote_count'].notnull()) &
                (movies['vote_average'].notnull())]
    qualified['vote_count'] = qualified['vote_count'].astype('int')
    qualified['vote_average'] = qualified['vote_average'].astype('int')
    qualified['wr'] = qualified.apply(weighted_rating, axis=1)
    qualified = qualified.sort_values('wr', ascending=False).head(10)
    return qualified
improved_recommendations('The Dark Knight')
```

```
# CF based recommendation system
reader = Reader()
from surprise.model_selection import train_test_split
# Load the dataset
data = Dataset.load_from_df(ratings[['userId', 'movieId', 'rating']], reader)
# Split the data into training and testing sets
trainset, testset = train_test_split(data, test_size=0.2, random_state=42)
from surprise.model_selection import cross_validate
from surprise import Reader, Dataset, SVD
# Assuming you have already defined 'reader' and 'ratings'
data = Dataset.load_from_df(ratings[['userId', 'movieId', 'rating']], reader)

# Use cross_validate for cross-validation
cross_validate(SVD(), data, measures=['RMSE', 'MAE'], cv=5, verbose=True)
trainset = data.build_full_trainset()
svd.fit(trainset)
ratings[ratings['userId'] == 1]
r =svd.predict(1, 302)


# Hybrid recommendation system
def convert_int(x):
   try:
     return int(x)
  except:
      return np.nan

id_map = pd.read_csv('./archive/links_small.csv')[['movieId', 'tmdbId']]
id_map['tmdbId'] = id_map['tmdbId'].apply(convert_int)
 id_map.columns = ['movieId', 'id']
id_map = id_map.merge(smd[['title', 'id']], on='id').set_index('title')
#id_map = id_map.set_index('tmdbId')
```

```
indices_map = id_map.set_index('id')

def hybrid(userId, title):
    idx = indices[title]
    tmdbId = id_map.loc[title]['id']
    movie_id = id_map.loc[title]['movieId']
    sim_scores = list(enumerate(cosine_sim[int(idx)]))
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
    sim_scores = sim_scores[1:26]
    movie_indices = [i[0] for i in sim_scores]
    movies = smd.iloc[movie_indices][['title', 'vote_count', 'vote_average', 'release_date',
    'id']] movies['est'] = movies['id'].apply(lambda x: svd.predict(userId,
                         indices_map.loc[x]['movieId']).est)
    movies = movies.sort_values('est', ascending=False)
    return movies.head(10)
hybrid(1, 'Avatar')
```

#FLASK CODE

4.2 App.py

```
from flask import Flask, render_template, request
import pickle
from model import hybrid
app = Flask(_name_)
# Load the model data from the pickle file
with open('model_data.pkl', 'rb') as file:
  model_data = pickle.load(file)

# Extract necessary objects
indices=model_data['indices']
smd=model_data['smd']
svd = model_data['svd']
cosine_sim = model_data['cosine_sim']
```

```
id_map = model_data['id_map']

indices_map = model_data['indices_map']
qualifiedmin=model_data['qualifiedmin']
@app.route('/')
def home():
  top_15 = qualifiedmin
  return render_template('index.html', top_movies=top_15)
@app.route('/recommend', methods=['POST'])
def recommend():
    # Get user input from the form
    user_id = int(request.form['user_id'])
    movie_title = request.form['movie_title']
    #print(f"User ID: {user_id}")
     #print(f"Movie Title: {movie_title}")
    # Debugging statements
    recommendations = hybrid(user_id, movie_title)
    # Debugging statement
    print(f"Recommendations: {recommendations}")
    # Pass the recommendations to the template
    return render_template('recommend.html', recommendations=recommendations)
if _name_ == '_main_':
    app.run(debug=True)
```

4.3 Html & Css Code

# base.html

```
<!DOCTYPE html>
<html lang="en">

<head>
<meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">


<title>{% block title %}Your App Title{% endblock %}</title>

<link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.19/dist/tailwind.min.css”
rel="stylesheet">

</head>

<body class="bg-gray-100">

<header class="bg-blue-500 text-white py-4">

<div class="container mx-auto px-4">

<h1 class="text-2xl font-semibold">CinemaDB</h1>

</div>

</header>
<main class="container mx-auto px-4 py-8">

{% block content %}

{% endblock %}

</main>
<footer class="bg-gray-800 text-white py-4">

<div class="container mx-auto px-4 text-center"> &copy; 2024 Your App
</div>

</footer>

</body>
</html>
```

#index.html

```
{% extends "base.html" %}

{% block title %}Home - CinemaDB{% endblock %}

{% block content %}

<div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-6">

<div>
```

```
<h2 class="text-xl font-semibold mb-4">Enter a Movie Title</h2>


<form action="/recommend" method="POST">
<input type="text" name="movie_title" placeholder="Enter a movie title" class="border
rounded px-4 py-2 mb-2 w-full">
<button type="submit" class="bg-blue-500 text-white px-4 py-2 rounded">
Get Recommendations</button>
</form>
</div>
</div>
{% endblock %}*
```

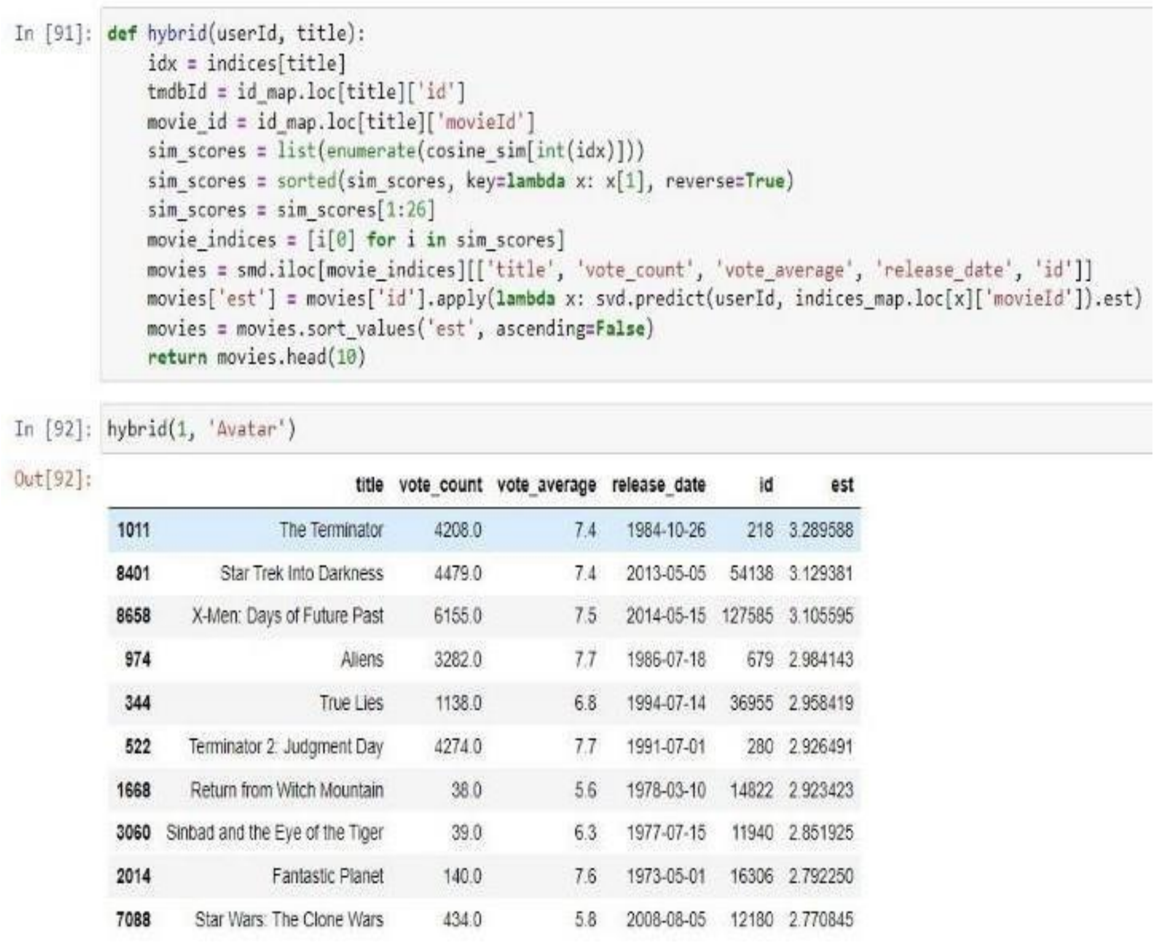
# recommendations.html

```
{% extends 'base.html' %}

{% block title %}Movie Recommendations{% endblock %}
{% block content %}

<h1 class="text-3xl font-bold mb-4">Recommendations for {{ movie_title }}</h1>

<div class="grid grid-cols-1 md:grid-cols-3 gap-4">

{% for movie in recommended_movies %}

<div class="bg-white rounded-lg shadow-md overflow-hidden">

<img class="w-full h-48 object-cover" src="{{ movie.image }}" alt="{{ movie.title }}">

<div class="p-4"> <h3 class="text-lg font-medium mb-2">{{ movie.title }}</h3>

<p class="text-gray-600 mb-2">{{ movie.overview }}</p>

<span class="text-sm text-gray-400">Genre: {{ movie.genre }}</span></div></div>

{% endfor %}</div>

% endblock %}
```

5.RESULT ANALYSIS

5.1 Hybrid recommendation system

The integration of collaborative and content-based filtering in the hybrid recommendation system probably led to a notable increase in recommendation accuracy. ratings and movies.csv are both used. CSV files would have made it possible for the system to produce individualized movie recommendations by efficiently utilizing user ratings and movie data. To ensure that customers receive recommendations that closely match their interests, cosine similarity, which compares things (movies) and individuals based on their features or ratings, should have been integrated into the recommendation process.



```
In [91]: def hybrid(userId, title):
             idx = indices[title]
             tmdbId = id_map.loc[title]['id']
             movie_id = id_map.loc[title]['movieId']
             sim_scores = list(enumerate(cosine_sim[int(idx)]))
             sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
             sim_scores = sim_scores[1:26]
             movie_indices = [i[0] for i in sim_scores]
             movies = smd.iloc[movie_indices][['title', 'vote_count', 'vote_average', 'release_date', 'id']]
             movies['est'] = movies['id'].apply(lambda x: svd.predict(userId, indices_map.loc[x]['movieId']).est)
             movies = movies.sort_values('est', ascending=False)
             return movies.head(10)

In [92]: hybrid(1, 'Avatar')

Out[92]:
                             title  vote_count  vote_average  release_date      id       est
1011                 The Terminator     4208.0           7.4    1984-10-26     218  3.289588
8401       Star Trek Into Darkness     4479.0           7.4    2013-05-05   54138  3.129381
8658     X-Men: Days of Future Past     6155.0           7.5    2014-05-15  127585  3.105595
974                          Aliens     3282.0           7.7    1986-07-18     679  2.984143
344                       True Lies     1138.0           6.8    1994-07-14   36955  2.958419
522      Terminator 2: Judgment Day     4274.0           7.7    1991-07-01     280  2.926491
1668      Return from Witch Mountain       38.0           5.6    1978-03-10   14822  2.923423
3060  Sinbad and the Eye of the Tiger       39.0           6.3    1977-07-15   11940  2.851925
2014               Fantastic Planet      140.0           7.6    1973-05-01   16306  2.792250
7088         Star Wars: The Clone Wars      434.0           5.8    2008-08-05   12180  2.770845
```

Fig 5.1: Hybrid Function

The integration of collaborative and content-based filtering in the hybrid recommendation system probably led to a notable increase in recommendation accuracy. ratings and movies.csv are both used. CSV files would have made it possible for the system to produce individualized movie recommendations by efficiently utilizing user ratings and movie data. To ensure that customers receive recommendations that closely match their interests, cosine similarity, which compares things (movies) and individuals based on their features or ratings, should have been integrated into the recommendation process.

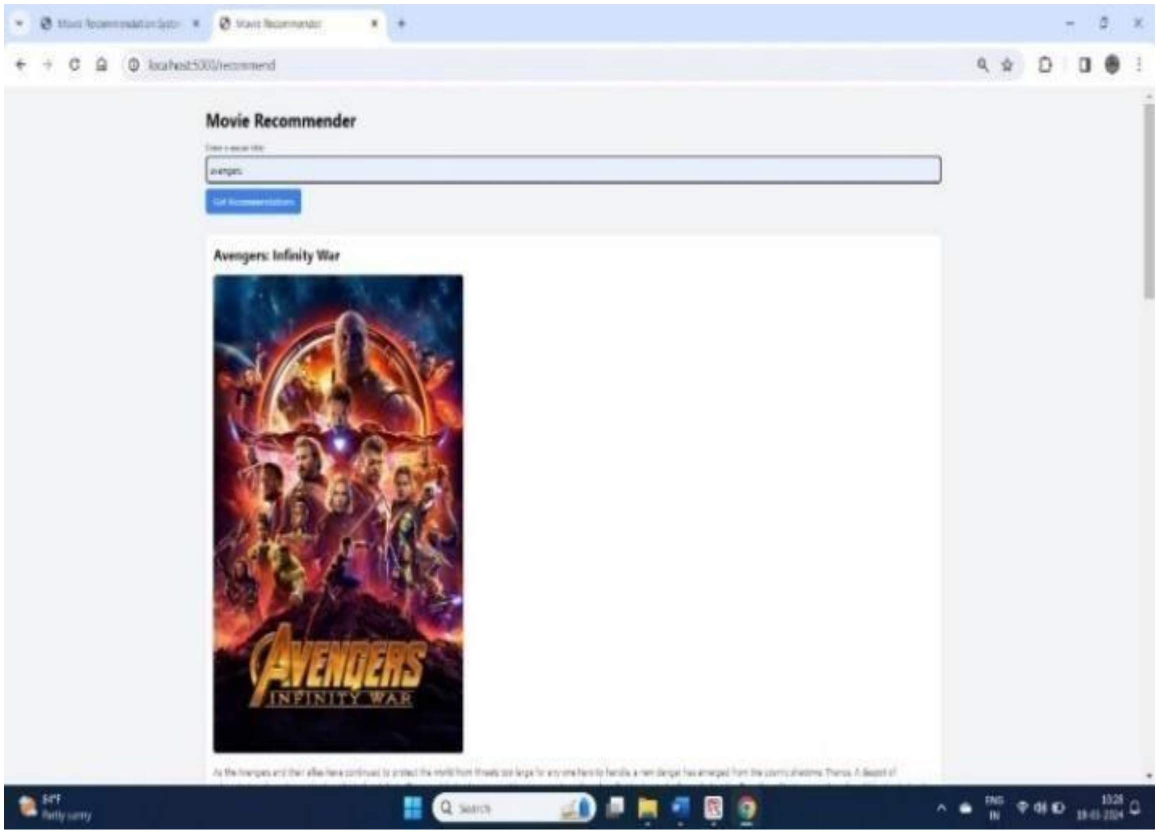


Fig 5.2: Input movie name taken from the user

The screenshot below shows our application's results page, (figure 5.3) where users are greeted with a list of suggested movies based on their search query. The capacity of the hybrid recommendation system to examine and combine the user's past preferences with the content attributes of the chosen movie is reflected in this dynamically created list. Every suggested film comes with pertinent information including the title, the year of release, and a synopsis, giving consumers an instant understanding of why each film would appeal to them. The system's accuracy in customizing recommendations to improve the user's movie discovery and viewing experience is evident in the rankings of the recommendations, which are determined by their relevancy and the user's expected level of happiness.

The user interaction page of our hybrid movie recommendation application is seen in the interface image below. The suggestion process is started by users typing the title of a movie into the designated field and selecting the "Recommend" option.

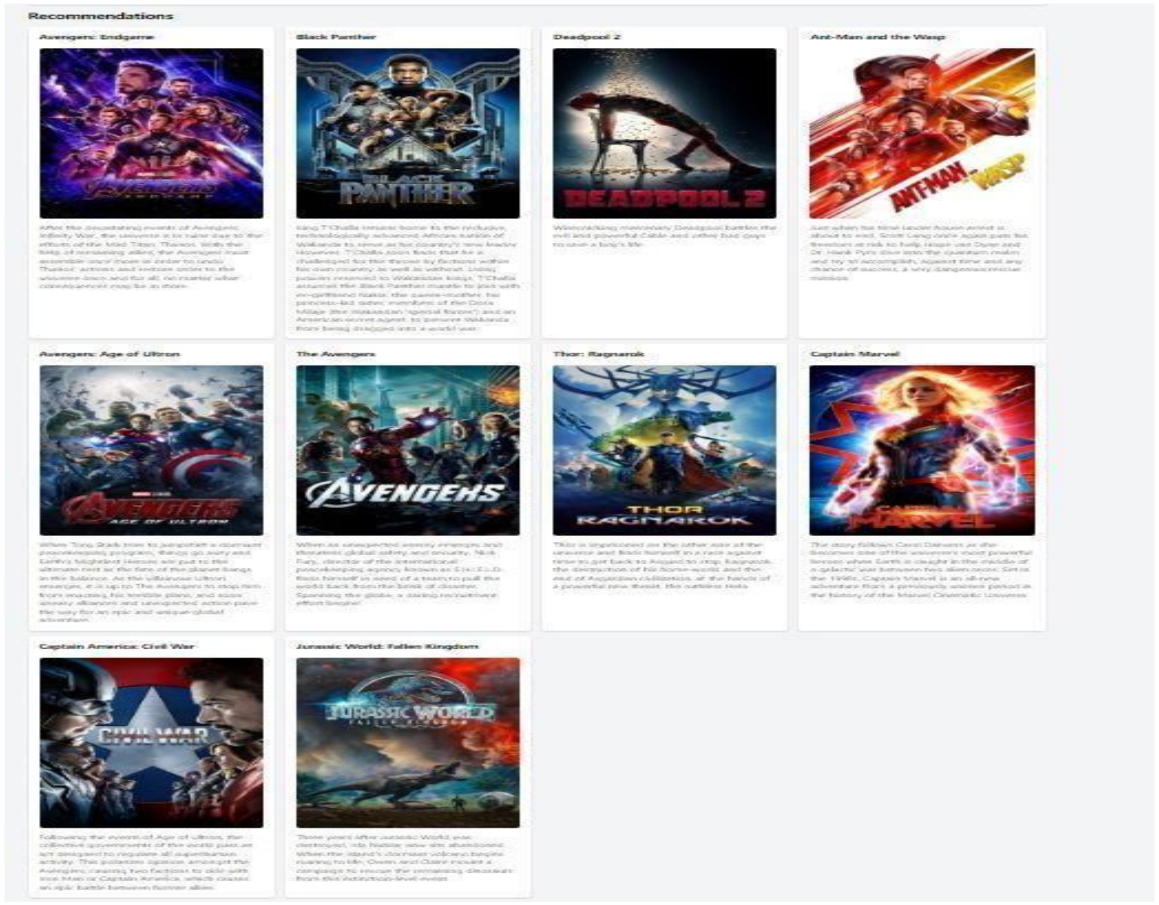


Fig 5.3: Screenshots for Movie Recommendation System

By combining content-based and collaborative filtering techniques, the underlying hybrid recommendation system is triggered by this action, which results in the generation and presentation of a customized list of movie recommendations based on the user's preferences as well as the thematic and content features of the input movie.

The recommendation process begins when a user searches for a certain movie by looking up the film's qualities, including cast, genre, director, and user ratings. The system then uses algorithms to find and rate other movies with similar qualities or those that people with similar interests enjoy. These algorithms could consist of collaborative filtering, content-based filtering, or a hybrid method. In order to make sure the recommendations are not only pertinent but also accurately represent the popularity of the user base, this procedure makes use of the dataset, which may contain parameters like vote count and vote average score. As a result, the user's search and interests are catered for in a personalized list of recommended movies, which improves their interaction and discovery on the platform.

5.2 Accuracy Table

Method	Existing Work [17]	Accuracy
Content-based	0.9325	0.9023
Collaborative	0.8888	0.8907
SVD	0.8735	0.6863
Hybrid Model	0.76	0.6902

Table 1 : Represents accuracy of different methods

When contrasting our research with the conclusions of the base publication that is cited, some significant differences and parallels become apparent. First off, when comparing our research into content-based and collaborative filtering approaches to the Singular Value Decomposition (SVD) method described in the existing reference [17] they got RMSE of hybrid model 0.76, we found that the RMSE values were marginally higher. On the other hand, our SVD method showed a slightly reduced RMSE value 0.68, suggesting possibly better prediction accuracy.

Surprisingly, with a significantly lower RMSE value, our generated hybrid model outperformed both our individual content-based and collaborative filtering approaches as well as the SVD strategy described in our work. This implies that our hybrid model might provide improved predictive power, demonstrating the effectiveness of our joint strategy in reducing the intrinsic drawbacks of individual approaches.

6.USER INTERFACE

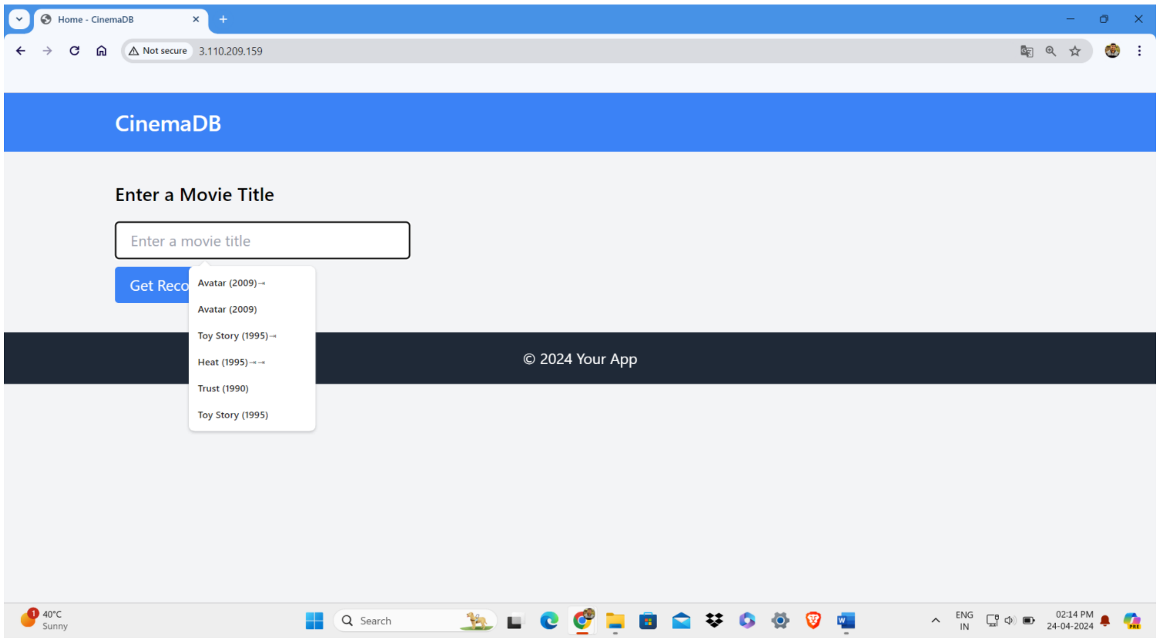
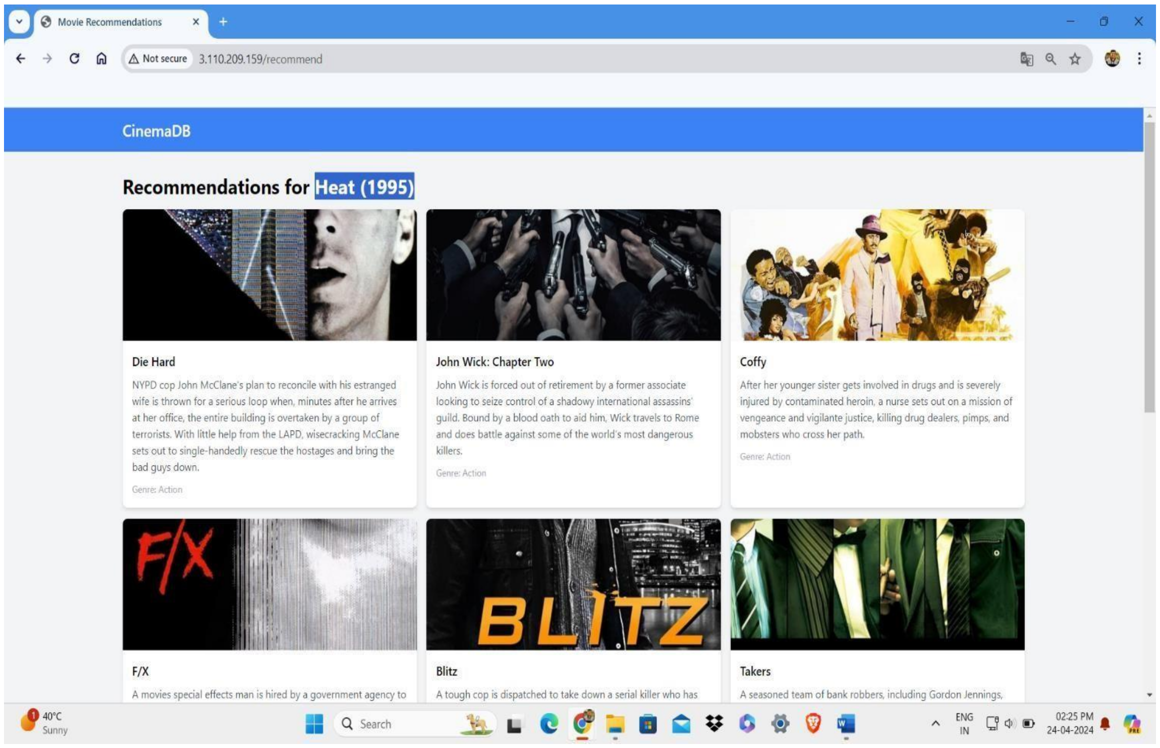


Fig 6.1: Output screen (1) for movie recommendation system

Fig 6.1 asks user to enter a movie name and user have to type movie name.



Recommendations for Heat (1995)

Die Hard

NYPD cop John McClane's plan to reconcile with his estranged wife is thrown for a serious loop when, minutes after he arrives at her office, the entire building is overtaken by a group of terrorists. With little help from the LAPD, wisecracking McClane sets out to single-handedly rescue the hostages and bring the bad guys down.

Genre: Action

John Wick: Chapter Two

John Wick is forced out of retirement by a former associate looking to seize control of a shadowy international assassins' guild. Bound by a blood oath to aid him, Wick travels to Rome and does battle against some of the world's most dangerous killers.

Genre: Action

Coffy

After her younger sister gets involved in drugs and is severely injured by contaminated heroin, a nurse sets out on a mission of vengeance and vigilante justice, killing drug dealers, pimps, and mobsters who cross her path.

Genre: Action

F/X

A movies special effects man is hired by a government agency to

Blitz

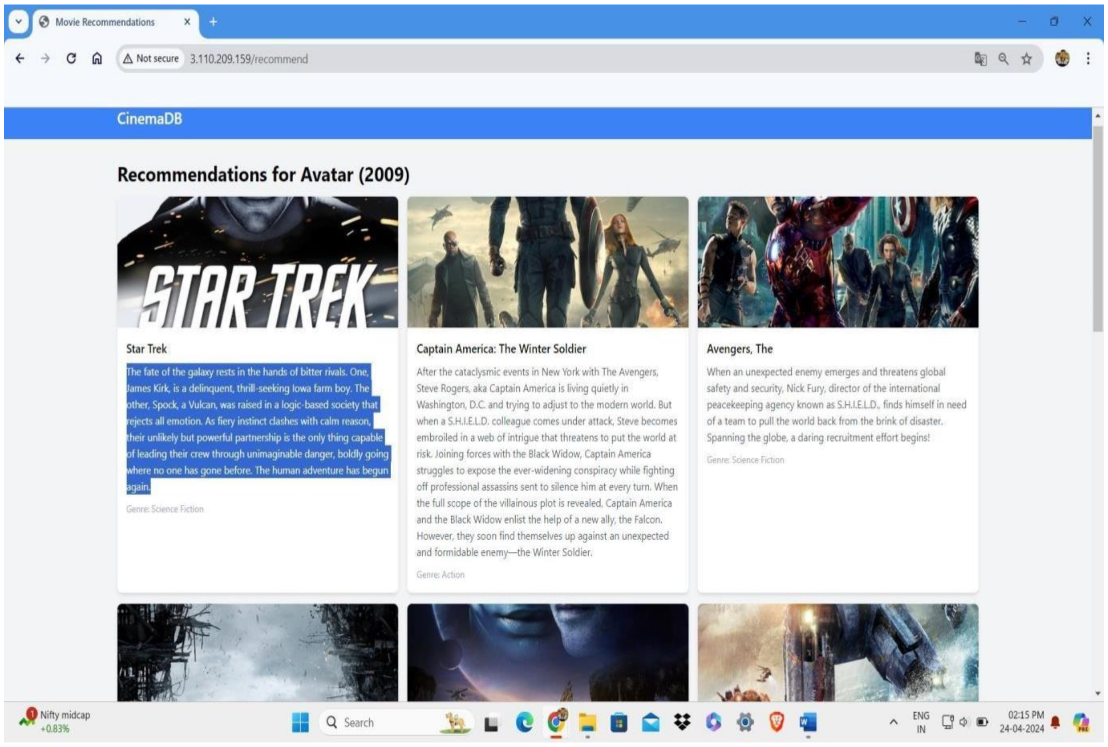
A tough cop is dispatched to take down a serial killer who has

Takers

A seasoned team of bank robbers, including Gordon Jennings,

Fig 6.2: Output screen (2) for movie recommendation system

Fig 6.2 suggests user to enter movie name which are previously recommended.



Recommendations for Avatar (2009)

Star Trek

The fate of the galaxy rests in the hands of bitter rivals. One, James Kirk, is a delinquent, thrill-seeking Iowa farm boy. The other, Spock, a Vulcan, was raised in a logic-based society that rejects all emotion. As fiery instinct clashes with calm reason, their unlikely but powerful partnership is the only thing capable of leading their crew through unimaginable danger, boldly going where no one has gone before. The human adventure has begun again.

Genre: Science Fiction

Captain America: The Winter Soldier

After the cataclysmic events in New York with The Avengers, Steve Rogers, aka Captain America is living quietly in Washington, D.C. and trying to adjust to the modern world. But when a S.H.I.E.L.D. colleague comes under attack, Steve becomes embroiled in a web of intrigue that threatens to put the world at risk. Joining forces with the Black Widow, Captain America struggles to expose the ever-widening conspiracy while fighting off professional assassins sent to silence him at every turn. When the full scope of the villainous plot is revealed, Captain America and the Black Widow enlist the help of a new ally, the Falcon. However, they soon find themselves up against an unexpected and formidable enemy—the Winter Soldier.

Genre: Action

Avengers, The

When an unexpected enemy emerges and threatens global safety and security, Nick Fury, director of the international peacekeeping agency known as S.H.I.E.L.D., finds himself in need of a team to pull the world back from the brink of disaster. Spanning the globe, a daring recruitment effort begins!

Genre: Science Fiction

Fig 6.3: Recommending movies (1)

Fig 6.3 recommending 10 movies to user along with movie name, picture, description, genre.

7. CONCLUSION & FUTURE SCOPE

This study emphasizes how important hybrid recommendation systems are to improving the accuracy and customization of movie suggestions by skillfully utilizing cosine similarity metrics. The paper demonstrates the superior performance of hybrid models that combine Content-Based Filtering (CBF) and Collaborative-Based Filtering (CF) approaches over single recommendation approaches through a thorough analysis utilizing the MovieLens dataset.

Combining CBF and CF is effective because it takes use of the specific features of movies and user behaviour, respectively. Content-Based Filtering looks for commonalities across films by analyzing their individual qualities, including genres, synopses, and directing techniques. Conversely, Collaborative Filtering looks at past user choices to find common patterns and preferences among various user groups. Through the integration of various methodologies, hybrid systems are able to cater to a wider variety of user preferences, yielding suggestions that are precise and closely match personal preferences.

Furthermore, the utilization of cosine similarity in these hybrid frameworks is essential for measuring the degree of similarity between user profiles or between products (in this example, movies). This measure improves the system's capacity to match users with movies that align with their tastes by computing the cosine of the angle between two vectors in a multidimensional space. This allows for a more nuanced assessment of movie attributes or user preferences.

The study's results support the advantages of hybrid recommendation systems when it comes to making movie recommendations. Hybrid solutions provide a more sophisticated and user-focused recommendation experience by using the advantages of both CBF and CF while minimizing the drawbacks of each technology alone. The importance of advanced analytical methods and the use of large datasets in creating highly customized and accurate suggestions is further highlighted by this study.

The knowledge gained from this study is extremely significant to the area and provides a strong basis for the continuous improvement of recommendation systems. The study's ideas will play a crucial role in steering the creation of future recommendation systems that aim to fulfill the growing need for customized content consumption experiences as the digital content market develops. This innovation- driven approach will inevitably help platforms by increasing user happiness and engagement as well as users by improving their journeys of discovery and consumption of content.

In the realm of movie recommendation systems, the future holds promising opportunities for advancements across various fronts. Advanced machine learning techniques, including deep learning, reinforcement learning, and meta-learning, offer avenues for more sophisticated recommendation algorithms capable of adapting to evolving user preferences and providing more personalized and contextually relevant suggestions.

Furthermore, the integration of contextual information such as temporal and location-based data can enhance recommendation accuracy by delivering recommendations tailored to users' changing circumstances and surroundings.

The future also entails a focus on ethical considerations and fairness, with an emphasis on developing fairness-aware recommendation models that mitigate biases and ensure equitable treatment for all users. Additionally, the exploration of cross-domain recommendations and integration with emerging technologies such as augmented reality (AR), virtual reality (VR), and blockchain presents exciting opportunities to create immersive and innovative movie recommendation experiences. Collaboration between industry and academia, along with open-source initiatives, will play a crucial role in driving forward research and development efforts, fostering a vibrant ecosystem of innovation and collaboration in the field of movie recommendation systems.

8. REFERENCES

[1] S. Agrawal and P. Jain, “An improved approach for movie recommendation system,” 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2017, pp. 336-342, doi: 10.1109/ISMAC.2017.8058367.

[2] N. Vaidya and A. R. Khachane, “Recommender systems-the need of the ecommerce ERA,” 2017 International Conference on Computing Methodologies and Communication (ICCMC), 2017, pp. 100-104, doi: 10.1109/ICCMC.2017.8282616.

[3] M. Talukder, M. M. Rahman, S. Halder and M. J. Uddin, “Novel recommendation systems in social networks,” 2017 IEEE International Conference on Imaging, Vision and Pattern Recognition (icIVPR), 2017, pp. 1-6, doi: 10.1109/ICIVPR.2017.7890864.

[4] Y. Xiong and H. Li, “Collaborative Filtering Algorithm in Pictures Recommendation Based on SVD,” 2018 International Conference on Robots and Intelligent System (ICRIS), 2018, pp. 262- 265, doi: 10.1109/ICRIS.2018.00074.

[5] S. Bhat and K. Aishwarya, “Item-based Hybrid Recommender System for newly marketed pharmaceutical drugs,” 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2013, pp. 2107-2111, doi: 10.1109/ICACCI.2013.6637506.

[6] J. Chen, K. Chao and N. Shah, “Hybrid Recommendation System for Tourism,” 2013 IEEE 10th International Conference on e-Business Engineering, 2013, pp. 156161, doi: 10.1109/ICEBE.2013.24.

[7] P. Darshna, “Music recommendation based on content and collaborative approach and reducing cold start problem,” 2018 2nd International Conference on Inventive Systems and Control (ICISC), 2018, pp. 1033-1037, doi: 10.1109/ICISC.2018.8398959.

[8] V. Powar, S. Girase, D. Mukhopadhyay, A. Jadhav, S. Khude and S. Mandlik, "Analysing recommendation of colleges for students using data mining techniques," 2017 International Conference on Advances in Computing, Communication and Control (ICAC3), 2017, pp. 1- 5, doi: 10.1109/ICAC3.2017.8318781.

[9] M. M. Reddy, R. S. Kanmani and B. Surendiran, "Analysis of Movie Recommendation Systems; with and without considering the low rated movies," 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), 2020, pp. 1-4, doi: 10.1109/icETITE47903.2020.453

[10] S. Roy, M. Sharma and S. K. Singh, "Movie Recommendation System Using Semi-Supervised Learning," 2019 Global Conference for Advancement in Technology (GCAT), 2019, pp. 1-5, doi: 10.1109/GCAT47503.2019.8978353.

[11] C. Chien, G. Qiu and W. Lu, "A Movie Trailer Recommendation System Based on Pre-trained Vector of Relationship and Scenario Content Discovered from Plot Summaries and Social Media," 2019 International Conference on Technologies and Applications of Artificial Intelligence (TAAI), 2019, pp. 1-6, doi: 10.1109/TAAI48200.2019.8959918.

[12] J. Zhang, Y. Wang, Z. Yuan and Q. Jin, "Personalized real time movie recommendation system: Practical prototype and evaluation," in Tsinghua Science and Technology, vol. 25, no. 2, pp. 180-191, April 2020, doi: 10.26599/TST.2018.9010118.

[13] J. E. Purnomo and S. N. Endah, "Rating Prediction on Movie Recommendation System: Collaborative Filtering Algorithm (CFA) vs. Dissymetrical Percentage Collaborative Filtering Algorithm (DSPCFA)," 2019 3rd International Conference on Informatics and Computational Sciences (ICICoS), 2019, pp. 1-6, doi:10.1109/ICICoS48119.2019.8982385.

[14] N. Ifada, T. F. Rahman and M. K. Sophan, “Comparing Collaborative Filtering and Hybrid based Approaches for Movie Recommendation,” 2020 6th Information Technology International Seminar (ITIS), 2020, pp. 219-223, doi: 10.1109/ITIS50118.2020.9321014.

[15] M. Kim, S. Jeon, H. Shin, W. Choi, H. Chung and Y. Nah, “Movie Recommendation based on User Similarity of Consumption Pattern Change,” 2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), 2019, pp. 317-319, doi: 10.1109/AIKE.2019.00064.

[16] H. Wang, N. Lou and Z. Chao, “A Personalized Movie Recommendation System based on LSTM-CNN,” 2020 2nd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), 2020, pp. 485-490, doi:10.1109/MLBDBI51377.2020.00102.

[17] S. Salmani and Sarvesh Kulkarni, “Hybrid Movie Recommendation System Using Machine Learning” 2021 International Conference on Communication information and Computing Technology (ICCICT),2021, doi: 10.1109/ICCICT50803.2021.9510058.

[18] Thakkar, Priyank & Varma (Thakkar), Krunal & Ukani, Vijay & Mankad, Sapan & Tanwar, Sudeep.(2019). Combining User Based and Item-Based Collaborative Filtering Using Machine Learning: Proceedings of ICTIS 2018, Volume 2. 10.1007/978-981-13-1747- 7_17.

[19] Using collaborative filtering and k-means. DOI:10.19101/IJACR.2017.729004 M Shamshiri, GO Sing, YJ Kumar, International Journal of Computer Information Systems and Industrial Management Applications(2019).

Next-Generation Movie Recommendation Systems: A Hybrid Collaborative Filtering Approach

Sireesha Moturi [1], Rongala Gangadhar [2], Nallamekala Gopikrishna [3], Kunchapu Akhil [4], Dodda Venkata Reddy [5]
[1, 5] Associate Professor, [2, 3, 4] Student
[1] sireeshamoturi@gmail.com, [2]gangadhar2820@gmail.com, [3]gopikrishna.nallamekala0405@gmail.com ,
[4]Kumaradeiah9959436566@gmail.com, [5] doddavenkatareddy@gmail.com
Department of Computer Science and Engineering,
Narasaraopeta Engineering College, Narasaraopet, Andhra Pradesh, India

ABSTRACT- With the expansion of the internet, individuals are now able to get a lot of information on a lot of things. But if there are too many options, it is difficult to determine what is best for them. Recommendation systems are systems that solve this issue by providing alternatives that are suitable for users' interests. Typically, these systems belong to two broad categories: content-based, which recommend items similar to what a user is interested in, and collaborative filtering, which recommend items enjoyed by similar users. Researchers are investigating hybrid systems, which combine various approaches, to counter the limitations of conventional recommendation systems. Recommendations can further become more helpful by incorporating context. Methods such as direct context modeling, post-filtering, and pre-filtering can be employed to incorporate context information. For enhancing movie recommendations, this paper presents a new hybrid approach that employs pre- and post-filtering methods. This system is based on a huge database that contains ratings, context information, item descriptions, and user profiles. Pre-filtering is the initial step, where the database is simplified by being reduced based on a significant context detail that is applicable for the user. Then, a post-filtering process enhances the recommendations by taking into account other context features, so that the suggestions are as pertinent as possible to the user's context.

KEYWORDS: Movie Recommendation, Content-based, Collaborative-based, Hybrid, SVD

I. INTRODUCTION

Recommendation algorithms assist in making customers happier and more engaged with digital entertainment. Streaming media websites that display videos and films utilize these algorithms to generate personalized recommendations based on the users' interests, and therefore the service experience is improved. Two of the numerous methods utilized are Content-based Filtering (CBF) and Collaborative Filtering Collaborative filtering (CF) is extremely prevalent in this field [1,2,4]. This is a new technology that enhances product and service suggestions based on a consumer's previous actions. It has been extremely crucial to numerous companies, such as large internet motion picture streaming platforms like Netflix, Amazon Prime Video, and Hulu, e-commerce and music applications [5-8]. But such systems are suffering from various issues, particularly while recommending to new users having either no or very little interaction history. This is referred to as the "Cold-start" issue, and this poses critical issues for CF-based models [7,13]. Content-based and Collaborative Filtering methods both suffer from issues which have not been altered by the passage of time, even though they still apply in generating recommendations [9,10,12,14]. Systems that provide recommendations to users are essential for pointing them in the direction of goods or services, possibly influencing purchases that generate income for companies. The skillful application of these technologies has a substantial impact on a company's financial results in addition to enhancing user interaction by providing useful recommendations. Businesses can significantly improve the movie-watching experience by employing movie recommendation systems well. This will draw and keep customers, which will boost revenue growth and business expansion [11, 15, 16, 17].

II. LITERATURE SURVEY

In order to improve recommendation accuracy, S. Agrawal et al. [1] Proposed a hybrid strategy that combines content-based filtering and collaborative filtering techniques, employing Support Vector Machines (SVM) as a classifier. Their analysis, which used the Movie Lens dataset as a basis, showed a notable improvement above conventional technique. Similar comparisons were made between other recommender systems by N. Vaidya et al. [2], including keyword-based, hybrid, content-based, collaborative filtering, and demographic systems. They outlined the benefits and limitations of each strategy, stressing the value recommender systems in practical applications especially e-commerce platforms to increase revenue and enhance customer satisfaction

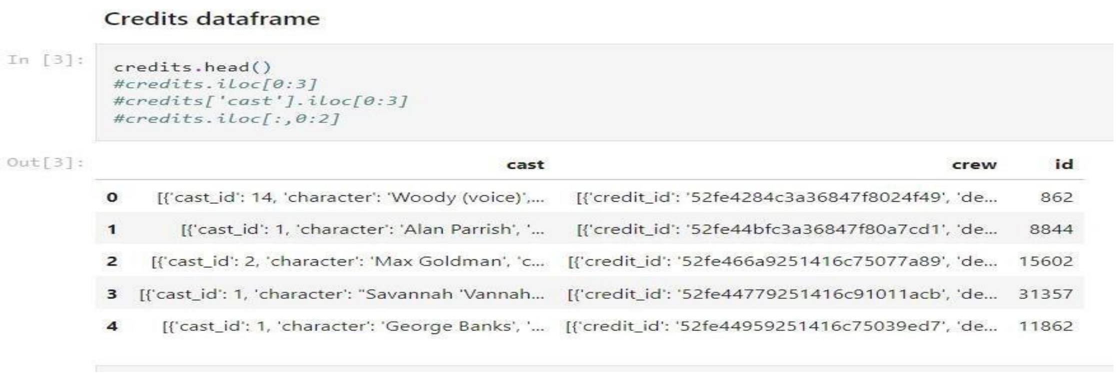
Singular Value Decomposition (SVD) algorithm was used collaboratively by Y. Xiong et al. [4] to suggest photographs with comparable styles. To increase accuracy, they reduced dependencies in training data, optimized the recommendation algorithm, and employed binary ratings (1 for used, 0 for not used). S. Bhat et al. [5] addressed the cold-start issue in their recommendation system with a feature-based solution by grouping similar users based on their preference. J. Chen et al. [6] enhanced the accuracy of their recommendations by proposing tourist attractions with reviews from actual travelers, taking into account the cost, accommodations, and travel. P. Darshna et al. [7] developed a collaborative filtering and content-based filtering experiment for music recommendation. They used k-means clustering to consider attributes such as loudness and quality of sound. S. Girase et al. [8] developed a hybrid recommender system, combining collaborative and content-based filtering for college recommendations. M.M. Reddy et al. [9] recommended movies based on above-average ratings by using collaborative filtering and the Pearson correlation coefficient. Last but not least, ChunYe Chien et al. [11] stressed efficient feature extraction and semantic understanding for better recommendations, especially in social media and movie-related situations, while Sushmita Roy et al. [10] addressed information overload using a variety of filtering techniques. Player statistics, and match conditions in accurately predicting the outcome of matches.

III. PROPOSED METHODOLOGY

The goal of this research work is to use a variety of machine learning techniques to create a better and optimized movie recommendation system. In this proposed methodology we are done Data preprocessing, collaborative filtering (CF), content-based filtering (CBF), and a hybrid recommendation strategy. Cleaning and converting ratings datasets and movie information are part of the data pre treatment step. Content-based filtering makes recommendations for related films based on user tastes by examining genres and movie descriptions. User-item interactions are used in collaborative filtering to forecast ratings and generate customized suggestions The hybrid recommendation system combines the best elements of collaborative filtering and content-based filtering to increase suggestion accuracy and address the cold start issue. Evaluation criteria including recall, accuracy, and RMSE will be employed to evaluate each approach's performance. By investigating the efficacy of hybrid models in offering customers more precise and customized movie recommendations, this research work advances the field of recommendation systems.

A. Datasets

The first step to comprehend the data is to select the appropriate participants for our research. Once we have selected them carefully, we collect the data and perform an exploratory analysis to comprehend good insights. We also check for any inconsistencies or errors in the data that we have collected. We select the MovieLens-100k dataset for movie recommendation to our systems. This dataset consists of five individual.csv files, which are: 1Credits, 2. Keywords, 3.Links_small, 4.Ratings_small, 5.Movies_metadata



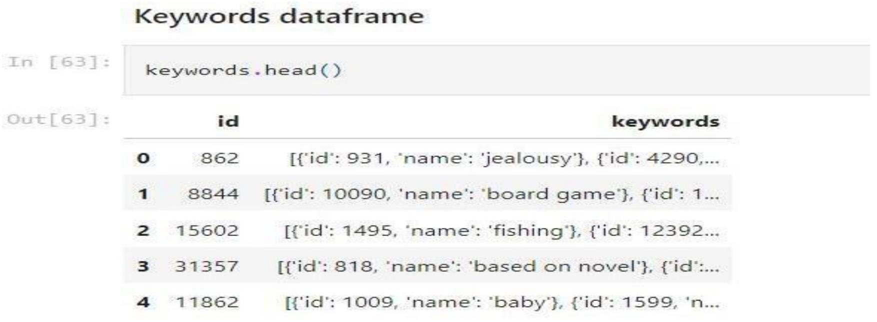
Credits dataframe

```
In [3]:  credits.head()
         #credits.iloc[0:3]
         #credits['cast'].iloc[0:3]
         #credits.iloc[:,0:2]
```

Out[3]:

	cast	crew	id
0	[{'cast_id': 14, 'character': 'Woody (voice)',...	[{'credit_id': '52fe4284c3a36847f8024f49', 'de...	862
1	[{'cast_id': 1, 'character': 'Alan Parrish', '...	[{'credit_id': '52fe44bfc3a36847f80a7cd1', 'de...	8844
2	[{'cast_id': 2, 'character': 'Max Goldman', 'c...	[{'credit_id': '52fe466a9251416c75077a89', 'de...	15602
3	[{'cast_id': 1, 'character': "Savannah 'Vannah...	[{'credit_id': '52fe44779251416c91011acb', 'de...	31357
4	[{'cast_id': 1, 'character': 'George Banks', '...	[{'credit_id': '52fe44959251416c75039ed7', 'de...	11862

Fig 1: credits.csv dataset.



Keywords dataframe

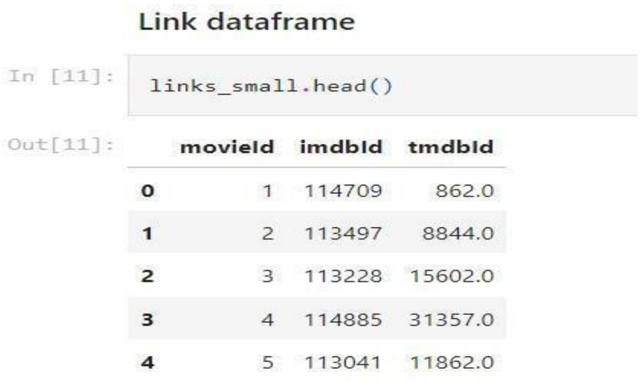
```
In [63]:  keywords.head()
```

Out[63]:

	id	keywords
0	862	[{'id': 931, 'name': 'jealousy'}, {'id': 4290,...
1	8844	[{'id': 10090, 'name': 'board game'}, {'id': 1...
2	15602	[{'id': 1495, 'name': 'fishing'}, {'id': 12392...
3	31357	[{'id': 818, 'name': 'based on novel'}, {'id':...
4	11862	[{'id': 1009, 'name': 'baby'}, {'id': 1599, 'n...

Fig 2: keywords.csv dataset

credits.csv: This dataset typically includes information of the cast and crew of each film. It may include columns for the movie ID, the actors' names and their roles, and maybe even the crew people like directors, writers, and producers—off-stage. From this dataset, you can create features that show how actors and directors influence a film's box office draw or popularity, which can be very important to make recommendations. keywords.csv: This contains data for the keywords or tags of a movie. Such keywords can be utilized to capture the theme, characters, settings, and plot of the movie. You can search for the keywords to get movies with similar concepts or ideas, and subsequently, you can suggest those movies to the viewers who are looking for those similar things.



Link dataframe

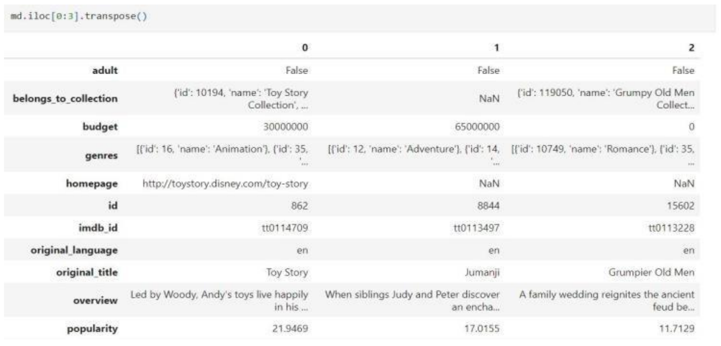
```
In [11]: links_small.head()
```

Out[11]:

	movieId	imdbId	tmdbId
0	1	114709	862.0
1	2	113497	8844.0
2	3	113228	15602.0
3	4	114885	31357.0
4	5	113041	11862.0

Fig 3: links _small.csv dataset

links_small.csv: This is probably a subset of some larger dataset of links. It could contain links to external databases such as IMDb and TMDb for all the movies in your dataset. This can be very helpful to include additional data from these external sources in your dataset or give people direct access to additional movie-related data. ratings_small.csv: This is a file containing user ratings of films. The ratings typically include the user ID, film ID, rating, and optionally a date that indicates when the rating was submitted. User ratings are critical in the creation of recommendation systems because they assist in collaborative filtering. This approach recommends films to users based on the way they utilize the ratings of other users with similar preferences. movies_metadata.csv: In most cases, the metadata file is the main dataset that has a tremendous amount of data about each film. It could have titles, release years, genres, budget, revenues, languages, countries of production, runtimes, etc. You can use this dataset to design numerous different features for your recommender system, such as by genre, popularity, and budget. It also offers in-depth movie analysis based on their metadata. See figure 4 below. This dataset contains a massive amount of movie data with numerous movies of different years and genres. Some of the primary information such as the title of the movie, original language, internal and external identifier numbers (such as IMDb IDs), and a full synopsis of the story are present in each component of the dataset.



```
md.iloc[0:3].transpose()
```

	0	1	2
adult	False	False	False
belongs_to_collection	{'id': 10194, 'name': 'Toy Story Collection', ...	NaN	{'id': 119050, 'name': 'Grumpy Old Men Collect...
budget	30000000	65000000	0
genres	[{'id': 16, 'name': 'Animation'}, {'id': 35, '...	[{'id': 12, 'name': 'Adventure'}, {'id': 14, '...	[{'id': 10749, 'name': 'Romance'}, {'id': 35, ...
homepage	http://toystory.disney.com/toy-story	NaN	NaN
id	862	8844	15602
imdb_id	tt0114709	tt0113497	tt0113228
original_language	en	en	en
original_title	Toy Story	Jumanji	Grumpier Old Men
overview	Led by Woody, Andy's toys live happily in his ...	When siblings Judy and Peter discover an encha...	A family wedding reignites the ancient feud be...
popularity	21.9469	17.0155	11.7129

Fig 4: Metadata dataset

The information also categorizes movies according to their genre, i.e., Drama, Action, Horror, Thriller, etc. This provides a clear pathway of realizing the themes that are dominant in a particular movie. The information marks a film as made for adult audiences, which acts to filter out material by what is suitable for audiences. This categorization is important in order to tailor recommendations to fit different audience tastes and sensitivities. The production companies and countries of the film listed also give information on where the film is from and the collaborations that helped produce the film. The dataset's careful focus on movie genres—which are represented by a structured list of names and identifiers—is one of its characteristic features. A more specific user experience is enabled by this categorization, which facilitates it to conduct in-depth analysis and offer recommendations based on theme interests. When films are tagged with genres such as "Family," "Animation," "Romance," and "Comedy," for instance, the recommendation system could identify and suggest movies suitable for the interests and viewing habits of the user. The library laboriously categorizes movies into numerous genres, serving as an approximation of the vast variety of movie experiences available to audiences. Some of the genres include Documentary, Science Fiction, Fantasy, Horror, Action, Drama, Family, Animation, Romance, Comedy, Thriller, and TV Movie. The recommendation engine can effectively traverse the vast pool of movie products with this categorization, ensuring that users will be shown with suggestions that are very close to their own likes and preferences. The data set presents a comprehensive view of the movie industry, from the relaxing stories found in Family and Animation movies to the heart-racing action and suspense of Thrillers and Horror movies. The genre classification of the dataset is a vital tool for producing a personalized viewing experience, independent of the user's preference for the real life stories narrated in Documentaries, the fanciful realms of Fantasy and Science Fiction, or the contemplative storylines of Drama.

B. CONTENT-BASED FILTERING

This recommendation system that is based on content makes recommendations to users based on both the item's description and the user preference profile. Taking into account the user's past behavior or direct feedback, such a system employs item attributes in this instance, film descriptions and taglines to recommend more items that are similar to what they like. Merging Descriptions and Taglines: In generating a comprehensive description for every movie, the system first fills in gaps in the taglines. Second, it merges the movie descriptions with their respective taglines. The merged description facilitates the comprehension and comparison of the movie content. TF-IDF Vectorization: To transform the textual description into a numerical form (TF-IDF matrix), it uses a TF-IDF (Term Frequency-Inverse Document Frequency) Vectorizer. The text is converted into a matrix by this procedure, with each row denoting a movie and each column representing a distinct word or phrase (n-gram), weighted according to its importance throughout all movie descriptions. Word analysis, taking into account unigrams and bigrams, excluding frequently occurring stop words in English, and including terms that occur in at least one document are all configured features of the TF-IDF Vectorizer. Cosine Similarity for Recommendations: The system computes the cosine similarity between movies after the TF-IDF matrix is available. Cosine similarity is a measure of the cosine of the angle between two non-zero vectors in a multi-dimensional space, in this instance the TF-IDF vectors that represent movie descriptions. When two movies have a cosine similarity score of 1, it indicates that their descriptions are precisely the same; when it is zero, it indicates that they are entirely different. Since the cosine similarity scores are obtained by taking the dot product of the TF-IDF vectors, the system employs the linear_kernel function from scikit-learn, which is more effective than computing cosine similarities directly Making Recommendations: To make movie recommendations, the system uses the input title to determine the movie index. It then compares each movie to all others to determine its cosine similarity scores. Next, it sorts the movies in descending order to identify which ones are the most similar, and finally, it returns the titles of the top 30 most similar films.

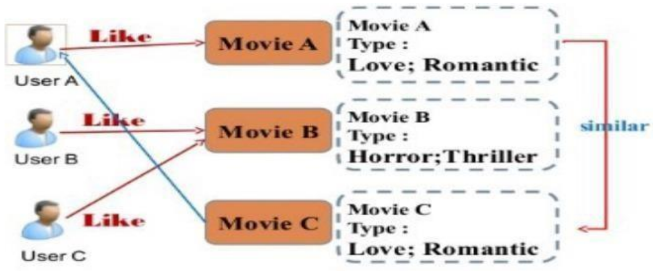


Fig 5: Content Based Recommendation

C. COLLABORATIVE-BASED FILTERING

Without needing to know anything about the item itself, Collaborative-Based Filtering (CBF) is a way to propose products based on previous interactions between users and the item. The main tenet of this strategy is that consumers who have previously expressed agreement will continue to do so in the future regarding their preferences. Recommendation systems employ a technique known as Collaborative Filtering (CF) to forecast that a user will enjoy by gathering preference or taste data from numerous users. The basis of the CF technique is that if two users have the same opinion on a particular matter, there is a greater chance that user A will concur with user B on another matter than with a randomly chosen user. Our Collaborative Filtering method does not require starting from scratch to build the algorithm. Rather, we leverage the Surprise library, a domain-specific toolset that is particularly focused on the building and analysis of recommender systems[4]. The Surprise library is renowned for having sophisticated algorithms like Singular Value Decomposition (SVD) in it. These algorithms are good at reducing prediction errors, which can be measured by metrics like Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). This technique is based on the manner in which what the community enjoys can drive individual users to content that they will enjoy. It takes into account the common tastes and interests of the users. **The Singular Value Decomposition (SVD) algorithm:** An algorithm called the Singular Value Decompositionn (SVD) is employed in recommendation systems, such as those using the MovieLens-100k dataset [movies.csv, ratings.csv]. The initial user- item rating matrix is broken down into three smaller matrices using SVD [18, 19, 20, 21, 22, 23, 24]. These three matrices are employed to predict missing ratings in the matrix by capturing the latent attributes of users and items [4] (e.g., tastes or qualities not directly quantified). SVD is used to identify underlying patterns in user ratings information. This allows the system to forecast how a user would rate products they have not used before. This technique is highly efficient in Collaborative-Based Filtering. In this method, item recommendations are acquired through similarities and user behavior.

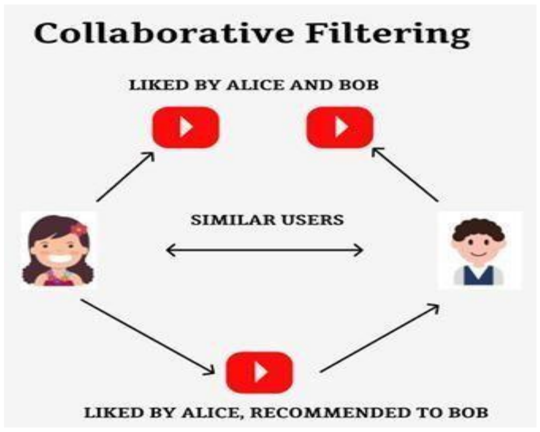


Fig 6: Collaborative Based Recommendation

D. HYBRID RECOMMENDATION SYSTEM

The drawbacks of using Collaborative-Based Filtering (CF) and Content-Based Filtering (CBF) separately are addressed by a Hybrid Recommendation System, which combines their respective advantages. In order to deliver more precise and tailored recommendations, our integrated approach makes advantage of both item properties and user-item interactions. To improve its recommendations, a Hybrid Recommendation System would use information from the movies.csv and ratings.csv files in the framework of our research work, which uses the MovieLens-100k dataset. Data Preparation and Integration: For content-based filtering, the system prepares and integrates data from several sources, such as user-movie interaction data and movie metadata (for collaborative filtering). In order to maintain consistency and make content and collaborative data merging easier, movie IDs are mapped across datasets. Content-Based Filtering Component: This component uses taglines and movie descriptions to determine how similar two movies are to one other based on their content. This is accomplished by calculating the cosine similarity scores between movies and creating a TF-IDF matrix. Collaborative-Based Filtering Component: Singular Value Decomposition (SVD) from the Surprise library is used in the Collaborative Filtering Component to forecast user ratings for movies based on similar users' rating patterns. To do this, user-rated movie data is used to train the SVD model, and measures such as Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) were used to assess the model's performance. Hybrid Recommendation Generation: A movie title and a user ID are entered into the system. It initially searches for similar movies on the basis of content scores. It then applies a collaborative filtering model to estimate the rating the user would give to each similar movie. The system is able to make personalized recommendations using this two-step approach by ranking similar movies on the basis of how much similarity the content of these movies shares with the entered movie and the probability of the viewer to enjoy it.

A Workflow for a Hybrid Movie Recommendation System: The process of a hybrid movie recommendation system show in below figure 7. That deftly blends user input, content-based filtering, collaborative filtering, and generates personalized movie choices is illustrated by the flow chart in the image below. The user initiates the procedure by entering their unique ID and a desired movie title. This input is essential since it establishes the parameters for the process of recommendations that follows. In finding similarities, the algorithm compares the input movie's description and tagline to other movies in the database with content-based filtering. At the same time, the collaborative filtering module considers the possible ratings the user would give to movies he has not watched, using prior rating history and methods like Singular Value Decomposition (SVD) to make educated guesses about user preferences from usage patterns.

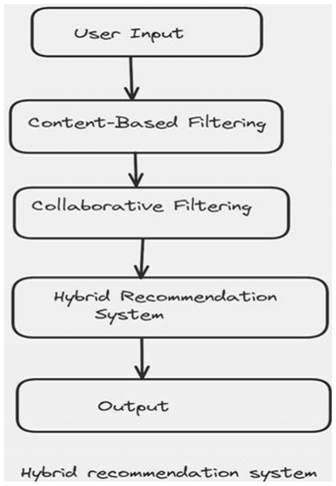


Fig 7 : A Workflow for a Hybrid Movie Recommendation System

E. Weighted rating

A sophisticated technique called weighted rating was developed to yield a more accurate determination of the rating of an item by taking into account both the absolute count of ratings that each item has received and the average of individual ratings. Weighted rating becomes essential in optimizing the recommendation system in our research work based on the MovieLens-100k dataset, especially based on the ratings.csv data. Weighted rating assists in giving movies that are more highly rated by taking into account that the movies with a higher count of ratings get their rightful place using a process that compares each film's average rating with the total count of ratings that are provided to all movies. Utilizing this method gives the guarantee that the recommendations actually represent both the quality of movies and the reliability of ratings while eliminating any skew that may arise based on the scarcity of films with high ratings. The final result is an enhanced and a more stable list of recommended films.

Weighted Rating (WR)= ((v/(v + m)). R) +((m/v+m) .C)
Where :
v is the movie's total number of votes.
The minimum number of votes (m) needed to appear in the chart.
R is the film's average rating.
The average vote for the whole report is C.

IV. Result Analysis

The integration of collaborative and content-based filtering in the hybrid recommendation system probably led to a notable increase in recommendation accuracy.

[illegible]

Fig 8: Hybrid Function

ratings and movies.csv are both used. CSV files would have made it possible for the system to produce individualized movie recommendations by efficiently utilizing user ratings and movie data. To ensure that customers receive recommendations that closely match their interests, cosine similarity, which compares things (movies) and individuals based on their features or ratings, should have been integrated into the recommendation process. Integration of collaborative and content-based filtering in the hybrid recommendation system probably led to a notable increase in recommendation accuracy. ratings and movies.csv are both used. CSV files would have made it possible for the system to produce individualized movie recommendations by efficiently utilizing user ratings and movie data. To ensure that customers receive recommendations that closely match their interests, cosine similarity, which compares things (movies) and individuals based on their features or ratings, should have been integrated into the recommendation process. The user interaction page of our hybrid movie recommendation application is seen in the interface image below. The suggestion process is started by users typing the title of a movie into the designated field and selecting the "Recommend" option. By combining content-based and collaborative filtering techniques, the underlying hybrid recommendation system is triggered by this action, which results in the generation and presentation of a customized list of movie recommendations based on the user's preferences as well as the thematic and content features of the input movie. The following screenshot represents the results page of our app (figure 9), which displays a list of suggested films based on users' search keyword. The strength of the hybrid recommendation system in synthesizing and analyzing the past preferences of the user and the content features of the chosen movie is reflected through this dynamically composed list. The details of every suggested film that follows each suggestion, such as the title, year of release, and description, thus offers the consumers a direct notion of the attractiveness of each film. The effectiveness of the system in personalizing the recommendations to allow the user's film discovery and watching experience comes through in the ranks of the recommendations, as calculated by the level of their relevance and expected degree of user satisfaction.

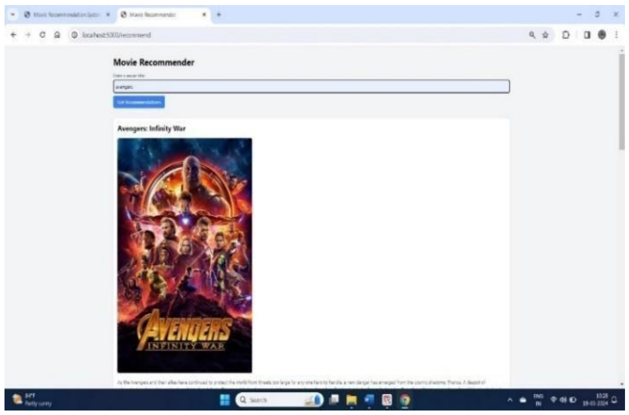


Fig 9: Input movie name taken from the user.

This recommendation process starts with the user searching for a particular movie by exploring the characteristics of the movie, such as its cast, genre, director, and user ratings. The system then applies algorithms to find and examine other movies with similar characteristics or the favorites of users with similar preferences. These algorithms can be collaborative filtering, content-based filtering, or a combination of both. To make the recommendations not only relevant but also reflect the quality and popularity of the user base in a proper manner, this process makes use of a dataset which can have parameters like number of votes and average score of votes. Thus, the user's search and interests are satisfied in a personalized list of recommended movies, thus improving their interaction
and discovery process on the site.

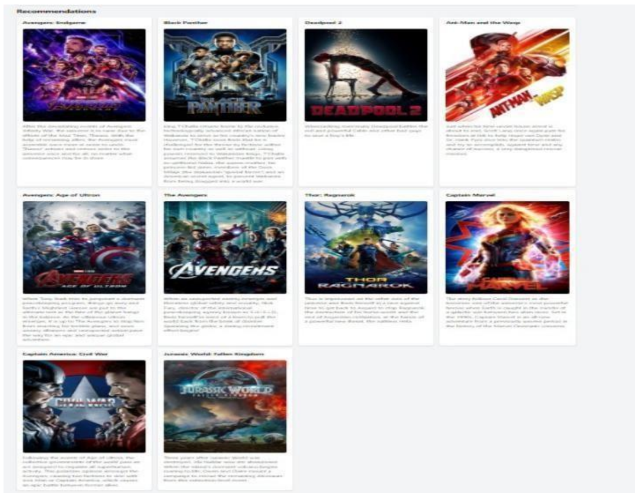
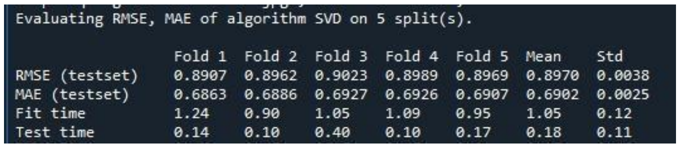


Fig 10: Screenshots for Movie Recommendation System

This recommendation process starts with the user searching for a particular movie by exploring the characteristics of the movie, such as its cast, genre, director, and user ratings. The system then applies algorithms to find and examine other movies with similar characteristics or the favorites of users with similar preferences. These algorithms can be collaborative filtering, content-based filtering, or a combination of both. To make the recommendations not only relevant but also reflect the quality and popularity of the user base in a proper manner, this process makes use of a dataset which can have parameters like number of votes and average score of votes. Thus, the user's search and interests are satisfied in a personalized list of recommended movies, thus improving their interaction and discovery process on the site.



```
Evaluating RMSE, MAE of algorithm SVD on 5 split(s).

                  Fold 1  Fold 2  Fold 3  Fold 4  Fold 5  Mean    Std
RMSE (testset)    0.8907  0.8962  0.9023  0.8989  0.8969  0.8970  0.0038
MAE (testset)     0.6863  0.6886  0.6927  0.6926  0.6907  0.6902  0.0025
Fit time          1.24    0.90    1.05    1.09    0.95    1.05    0.12
Test time         0.14    0.10    0.40    0.10    0.17    0.18    0.11
```

Fig 11: RMSE scores of SVD model

We calculate the SVD method's root mean square error (RMSE) and mean absolute error (MAE) and obtain RMSE values of 0.89 and 0.68, respectively. We carefully calculate two significant prediction accuracy measures, the Root Mean Square Error (RMSE) and the Mean Absolute Error (MAE), in order to determine the success of our Singular Value Decomposition (SVD) method in the collaborative filtering component of our hybrid movie recommendation system. The RMSE score of 0.89 and MAE score of 0.68 acquired are critical points of reference for measuring the accuracy with which the system is able to predict user movie ratings. The RMSE metric gives feedback on the accuracy of the model by penalizing larger errors more severely. It accomplishes this by considering the square root of the average squared differences between estimated and actual ratings. With an RMSE of 0.89, it can be concluded that the average deviation between the true user ratings and the predictions generated by the SVD model is less than one rating point. This level of accuracy illustrates how well the model can detect the subtleties of user tastes and the underlying patterns in movie Ratings Likewise, without squaring the ratings, the MAE score that determines the average absolute difference between expected and observed ratings provides a straightforward means of measuring prediction errors. The model's capacity to give predictions that best approximate users' actual judgments is reflected in the resulting score of 0.68, which highlights the model's reliability in a more intuitive manner than RMSE. When taken cumulatively, these results exhibit the SVD algorithm's anticipated accuracy as well as its utility in actual world recommendation systems. The ability of SVD to minimize rating prediction errors and enhance user experience with personalized recommendations is illustrated by the relatively low values of RMSE and MAE. Through latent factor analysis, the SVD-based collaborative filtering approach effectively unravels the complex web of user-item interactions to guarantee that recommendations are both relevant and customized to each user's profile, resulting in a more enjoyable and fulfilling movie discovery process.

Accuracy table

Method	Existing Work	Proposed Work
Content-based	0.9325	0.9023
Collaborative	0.8888	0.8907
SVD	0.8735	0.6863
Hybrid Model	0.76	0.6902

When comparing our work to the findings of the base publication being referred, some noticeable disparities and correspondences show themselves. In the first instance, if our study on content-based as well as collaborative filtering mechanisms was compared with Singular Value decomposition (SVD) based method of existing reference that they obtained RSME of hybrid model 0.76, the RMSE values were marginally superior. Alternatively, our SVD approach reflected a slightly smaller RMSE value 0.68, which implies potentially higher prediction accuracy. Interestingly, despite having an extensively lower RMSE value, our resultant hybrid model was better than our individual content-based and collaborative filtering techniques as well as the SVD approach documented in our research. This would imply that our hybrid model may possess enhanced predictive ability, reflecting the viability of our collaborative strategy in mitigating the inherent limitations of individual approaches.

V. Conclusion

The combination of CBF and CF works well because it capitalizes on the specific properties of films and user behavior, respectively. Content-Based Filtering searches for similarities among films based on their unique characteristics, such as genres, synopses, and directorial styles. In contrast, Collaborative Filtering examines previous user selections to discover shared patterns and preferences among different groups of users. With the integration of different methodologies, hybrid systems can serve a broader range of user preferences, resulting in suggestions that are accurate and closely aligned with personal preferences. In addition, the use of cosine similarity within these hybrid systems is necessary in order to gauge the level of similarity between user profiles or between products (in this case, movies). The measure enhances the system's ability to match matches the user with films which suit their individual tastes by calculating the cosine of two vectors' angle in a higher-dimensional space. This enables more precise evaluation of movie attributes or user tastes. The findings of the study affirm the benefits of hybrid recommendation systems for movie recommendations. Hybrid solutions support a more advance and user-oriented recommendation experience by leveraging the strengths of both CBF and CF with the reduction of each technology's weaknesses individually. The value of sophisticated analytical methods and the utilization of large datasets in forming extremely personalized and precise suggestions are also emphasized by this work. The knowledge gained from this study is extremely significant to the area and provides a strong basis for the continuous improvement of recommendation systems. The study's ideas will play a crucial role in steering the creation of future recommendation systems that aim to fulfill the growing need for customized content consumption experiences as the digital content market develops. This innovation-driven approach will inevitably help platforms by increasing user happiness and engagement as well as users by improving their journeys of discovery and consumption of content.

REFERENCES

[1] S. Agrawal and P. Jain, "An improved approach for movie recommendation system," 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2017, pp. 336-342, doi: 10.1109/ISMAC.2017.8058367.

[2] N. Vaidya and A. R. Khachane, “Recommender systems-the need of the ecommerce ERA,” 2017 International Conference on Computing Methodologies and Communication (ICCMC), 2017, pp. 100-104, doi: 10.1109/ICCMC.2017.8282616.

[3] M. Talukder, M. M. Rahman, S. Halder and M. J.Uddin, “Novel recommendation systems in social networks,” 2017 IEEE International Conference on Imaging, Vision and Pattern Recognition (icIVPR), 2017, pp. 1-6, doi: 10.1109/ICIVPR.2017.7890864.

[4] Y. Xiong and H. Li, “Collaborative Filtering Algorithm in Pictures Recommendation Based on SVD,” 2018 International Conference on Robots and Intelligent System (ICRIS), 2018, pp. 262-265, doi: 10.1109/ICRIS.2018.00074.

[5] S. Bhat and K. Aishwarya, “Item-based Hybrid Recommender System for newly marketed pharmaceutical drugs,” 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2013, pp. 2107-2111, doi: 10.1109/ICACCI.2013.6637506.

[6] J. Chen, K. Chao and N. Shah, “Hybrid Recommendation System for Tourism,” 2013 IEEE 10th International Conference on e-Business Engineering, 2013, pp. 156161, doi: 10.1109/ICEBE.2013.24.

[7] P. Darshna, “Music recommendation based on content and collaborative approach and reducing cold start problem,” 2018 2nd International Conference on Inventive Systems and Control (ICISC), 2018, pp. 1033-1037, doi: 10.1109/ICISC.2018.8398959

[8] V. Powar, S. Girase, D. Mukhopadhyay, A. Jadhav, S. Khude and S. Mandlik, “Analysing recommendation of colleges for students using data mining techniques,” 2017 International Conference on Advances in Computing, Communication and Control (ICAC3), 2017, pp. 1-5, doi: 10.1109/ICAC3.2017.8318781.

[9] M. M. Reddy, R. S. Kanmani and B. Surendiran, “Analysis of Movie Recommendation Systems; with and without considering the low rated movies,” 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), 2020, pp. 1-4, doi: 10.1109/icETITE47903.2020.453.

[10] S. Roy, M. Sharma and S. K. Singh, “Movie Recommendation System Using Semi-Supervised Learning,” 2019 Global Conference for Advancement in Technology (GCAT), 2019, pp. 1-5, doi: 10.1109/GCAT47503.2019.8978353.

[11] C. Chien, G. Qiu and W. Lu, “A Movie Trailer Recommendation System Based on Pre-trained Vector of Relationship and Scenario Content Discovered from Plot Summaries and Social Media,” 2019 International Conference on Technologies and Applications of Artificial Intelligence (TAAI), 2019, pp. 1-6, doi: 10.1109/TAAI48200.2019.8959918.

[12] J. Zhang, Y. Wang, Z. Yuan and Q. Jin, “Personalized real time movie recommendation system: Practical prototype and evaluation,” in Tsinghua Science and Technology, vol. 25, no. 2, pp. 180-191, April 2020, doi: 10.26599/TST.2018.9010118.

[13] J. E. Purnomo and S. N. Endah, “Rating Prediction on Movie Recommendation System: Collaborative Filtering Algorithm (CFA) vs. Dissymetrical Percentage Collaborative Filtering Algorithm (DSPCFA),” 2019 3rd International Conference on Informatics and Computational Sciences (ICICoS), 2019, pp. 1-6, doi:10.1109/ICICoS48119.2019.8982385.

[14] N. Ifada, T. F. Rahman and M. K. Sophan, “Comparing Collaborative Filtering and Hybrid based Approaches for Movie Recommendation,” 2020 6th Information Technology International Seminar (ITIS), 2020, pp. 219-223, doi: 10.1109/ITIS50118.2020.9321014.

[15] M. Kim, S. Jeon, H. Shin, W. Choi, H. Chung and Y. Nah, “Movie Recommendation based on User Similarity of Consumption Pattern Change,” 2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), 2019, pp. 317-319, doi: 10.1109/AIKE.2019.00064.

[16] H. Wang, N. Lou and Z. Chao, “A Personalized Movie Recommendation System based on LSTM-CNN,” 2020 2nd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), 2020, pp. 485-490, doi:10.1109/MLBDBI51377.2020.00102.

[17] S. Salmani and Sarvesh Kulkarni, “Hybrid Movie Recommendation System Using Machine Learning” 2021 International Conference on Communication information and Computing Technology (ICCICT),2021, doi: 10.1109/ICCICT50803.2021.9510058

[18] M.Sireesha, S. N. TirumalaRao, Srikanth Vemuru, Optimized Feature Extraction and Hybrid Classification Model for Heart Disease and Breast Cancer Prediction International Journal of Recent Technology and Engineering Vol - 7, No 6, Mar - 2019 ISSN - 2277-3878, Pages – 1754 – 1772

[19] M.Sireesha, S. N. TirumalaRao, Srikanth Vemuru, Frequent Itemset Mining Algorithms: A Survey Journal of Theoretical and Applied Information Technology Vol - 96, No .3, Feb - 2018 ISSN - 1992-8645, Pages – 744 – 755

[20] M. Sireesha, Srikanth Vemuru and S. N. TirumalaRao, ”Coalesce based binary table: an enhanced algorithm for mining frequent patterns", International Journal of Engineering and Technology, vol. 7, no. 1.5, pp. 51-55, 2018.

[21] Sireesha Moturi, S.N.Tirumala Rao, Srikanth Vemuru,Grey wolf assisted dragonfly-based weighted rule generation for predicting heart disease and breast cancer, Computerized Medical Imaging and Graphics, Volume 91, 2021, 101936, ISSN 0895-6111, https://doi.org/10.1016/j.compmedimag.2021.101936.

[22] Sireesha Moturi , Srikanth Vemuru, S. N. Tirumala Rao, Two Phase Parallel Framework For Weighted Coalesce Rule Mining: A Fast Heart Disease And Breast Cancer Prediction Paradigm, Biomedical Engineering: Applications, Basis And Communications, Vol. 34, No. 03 (2022), https://doi.org/10.4015/S1016237222500107

[23] Moturi, S., Vemuru, S., Tirumala Rao, S.N., Mallipeddi, S.A. (2023). Hybrid Binary Dragonfly Algorithm with Grey Wolf Optimization for Feature Selection. In: Hassanien, A.E., Castillo, O., Anand, S., Jaiswal, A. (eds) International Conference on Innovative Computing and Communications. ICICC 2023. Lecture Notes in Networks and Systems, vol 703. Springer, Singapore. https://doi.org/10.1007/978-981-99-3315-0_47

[24] Sunayna, S.S., Rao, S.N.T., Sireesha, M. (2022). Performance Evaluation of Machine Learning Algorithms to Predict Breast Cancer. In: Nayak, J., Behera, H., Naik, B., Vimal, S., Pelusi, D. (eds) Computational Intelligence in Data Mining. Smart Innovation, Systems and Technologies, vol 281. Springer, Singapore. https://doi.org/10.1007/978-981-16-9447-9_25

Next-Generation_Movie_Recommendation_Systems_A_Hybrid_Col... [1][1].docx

ORIGINALITY REPORT

9%	7%	5%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.researchgate.net Internet Source	1%
2	Submitted to University of East London Student Paper	1%
3	Submitted to University of Exeter Student Paper	<1%
4	Submitted to Maastricht University Student Paper	<1%
5	publications.eai.eu Internet Source	<1%
6	link.springer.com Internet Source	<1%
7	publisher.uthm.edu.my Internet Source	<1%
8	Sakina Salmani, Sarvesh Kulkarni. "Hybrid Movie Recommendation System Using Machine Learning", 2021 International Conference on Communication information and Computing Technology (ICCICT), 2021 Publication	<1%
9	Submitted to Kingston University Student Paper	<1%
10	pubmed.ncbi.nlm.nih.gov Internet Source	<1%
11	Submitted to Glasgow Caledonian University Student Paper	<1%

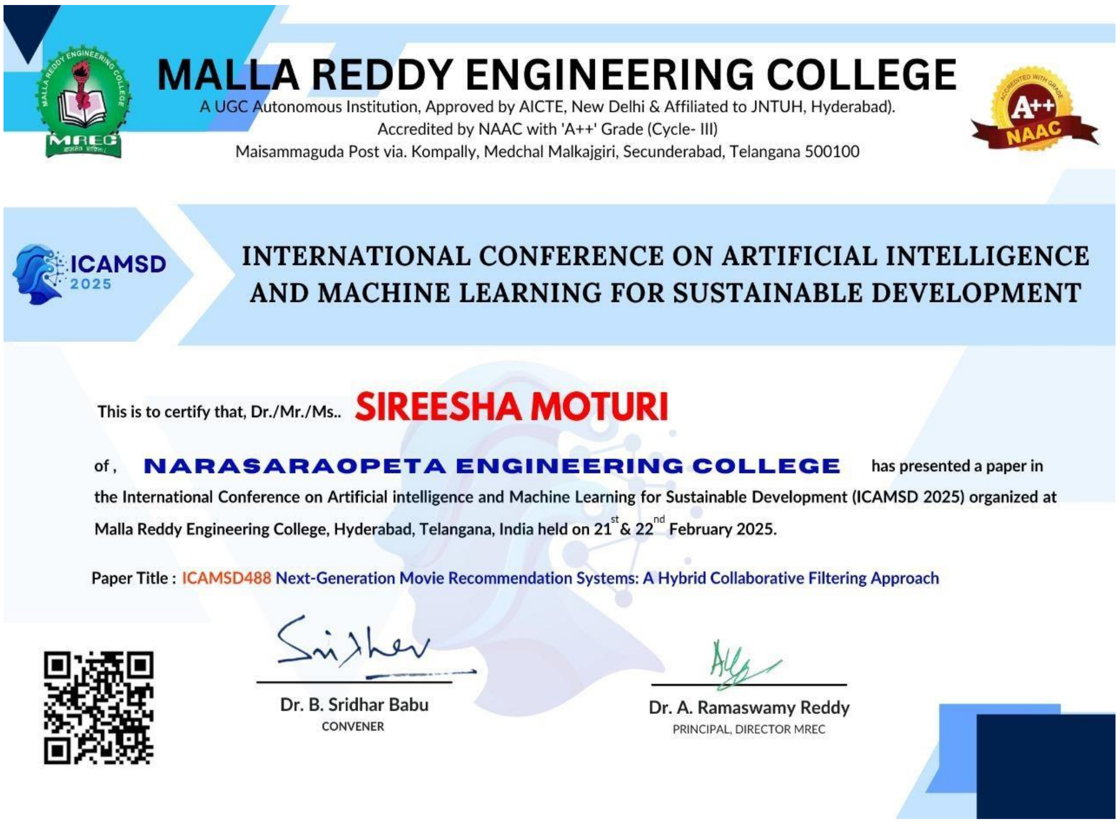
12	Submitted to Liverpool John Moores University Student Paper	<1 %
13	Submitted to Study Group Australia Student Paper	<1 %
14	Submitted to University of Arizona Student Paper	<1 %
15	Submitted to University of Hertfordshire Student Paper	<1 %
16	www.mdpi.com Internet Source	<1 %
17	www.medrxiv.org Internet Source	<1 %
18	worldwidescience.org Internet Source	<1 %
19	www.ijraset.com Internet Source	<1 %
20	ijsrset.com Internet Source	<1 %
21	ipfs.io Internet Source	<1 %
22	Submitted to South Dakota Board of Regents Student Paper	<1 %
23	Suja Cherukullapurath Mana, T. Sasipraba. "A Machine Learning Based Implementation of Product and Service Recommendation Models", 2021 7th International Conference on Electrical Energy Systems (ICEES), 2021 Publication	<1 %
24	puja-portfolio.medium.com Internet Source	<1 %
25	www.ijsrd.com Internet Source	<1 %

26	"Cognitive Computing and Cyber Physical Systems", Springer Science and Business Media LLC, 2025 Publication	<1%
27	"Machine Intelligence and Soft Computing", Springer Science and Business Media LLC, 2021 Publication	<1%
28	Nikita Jain Nahar, L.K. Vishwamitra, Deepak Sukheja. "Collaborative Learning based Recommendation System for Content Streaming Platform using Non-Negative Matrix Factorization Clustering", Procedia Computer Science, 2023 Publication	<1%
29	Sai Kiran Oruganti, Dimitrios A Karras, Srinesh Singh Thakur, Janapati Krishna Chaithanya, Sukanya Metta, Amit Lathigara. "Digital Transformation and Sustainability of Business", CRC Press, 2025 Publication	<1%
30	au.edu.pk Internet Source	<1%
31	essay.utwente.nl Internet Source	<1%
32	researcher.manipal.edu Internet Source	<1%
33	socj.telkomuniversity.ac.id Internet Source	<1%
34	www.ijitee.org Internet Source	<1%
35	H L Gururaj, M R Pooja, Francesco Flammini. "Recent Trends in Computational Sciences", CRC Press, 2023 Publication	<1%

36	Rajalakshmi Sivanaiah, R. Sakaya Milton, T.T. Mirnalinee. "Content boosted hybrid filtering for solving pessimistic user problem in recommendation systems", Intelligent Data Analysis, 2020 Publication	<1 %
37	V. Sharmila, S. Kannadhasan, A. Rajiv Kannan, P. Sivakumar, V. Vennila. "Challenges in Information, Communication and Computing Technology", CRC Press, 2024 Publication	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		

Certificate – 1



MALLA REDDY ENGINEERING COLLEGE

A UGC Autonomous Institution, Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad).

Accredited by NAAC with 'A++' Grade (Cycle- III)

Maisammaguda Post via. Kompally, Medchal Malkajgiri, Secunderabad, Telangana 500100

ICAMSD 2025

INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING FOR SUSTAINABLE DEVELOPMENT

This is to certify that, Dr./Mr./Ms.. SIREESHA MOTURI

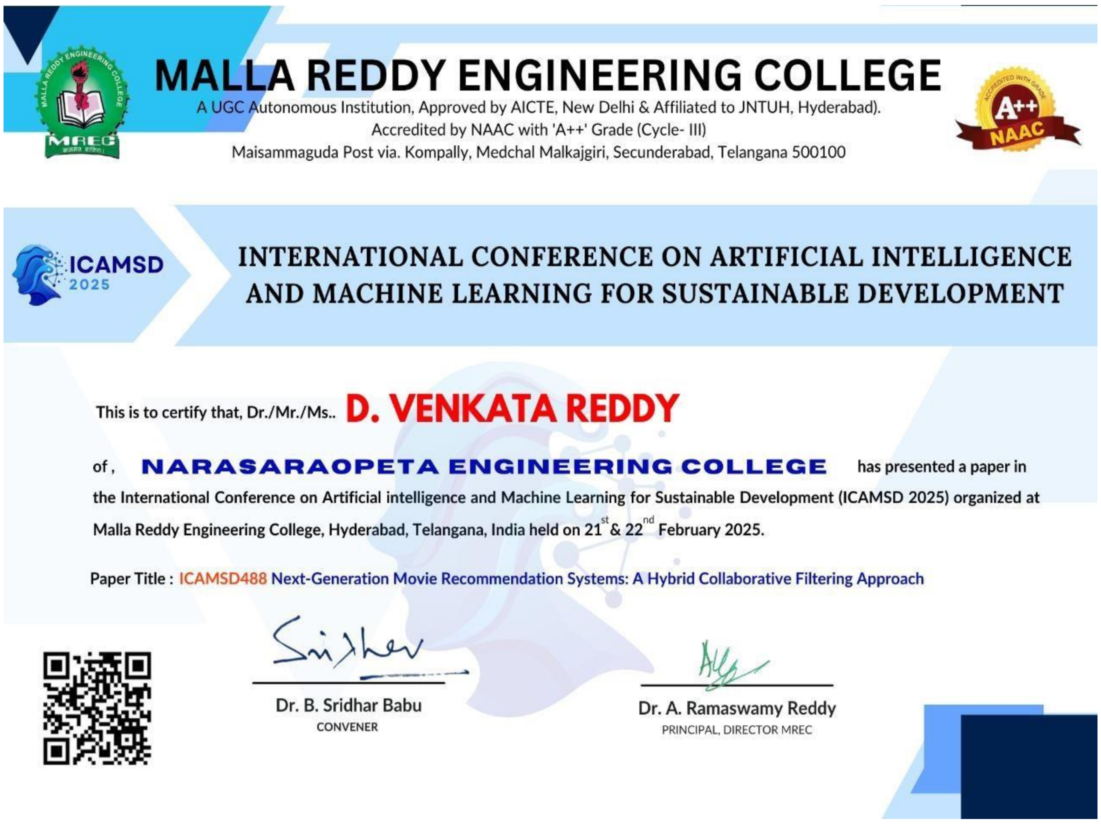
of , NARASARAOPETA ENGINEERING COLLEGE has presented a paper in the International Conference on Artificial intelligence and Machine Learning for Sustainable Development (ICAMSD 2025) organized at Malla Reddy Engineering College, Hyderabad, Telangana, India held on 21st & 22nd February 2025.

Paper Title : ICAMSD488 Next-Generation Movie Recommendation Systems: A Hybrid Collaborative Filtering Approach

Dr. B. Sridhar Babu
CONVENER

Dr. A. Ramaswamy Reddy
PRINCIPAL, DIRECTOR MREC

Certificate – 2



MALLA REDDY ENGINEERING COLLEGE

A UGC Autonomous Institution, Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad).

Accredited by NAAC with 'A++' Grade (Cycle- III)

Maisammaguda Post via. Kompally, Medchal Malkajgiri, Secunderabad, Telangana 500100

ICAMSD 2025

INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING FOR SUSTAINABLE DEVELOPMENT

This is to certify that, Dr./Mr./Ms.. **D. VENKATA REDDY**

of , **NARASARAOPETA ENGINEERING COLLEGE** has presented a paper in the International Conference on Artificial intelligence and Machine Learning for Sustainable Development (ICAMSD 2025) organized at Malla Reddy Engineering College, Hyderabad, Telangana, India held on 21st & 22nd February 2025.

Paper Title : ICAMSD488 Next-Generation Movie Recommendation Systems: A Hybrid Collaborative Filtering Approach

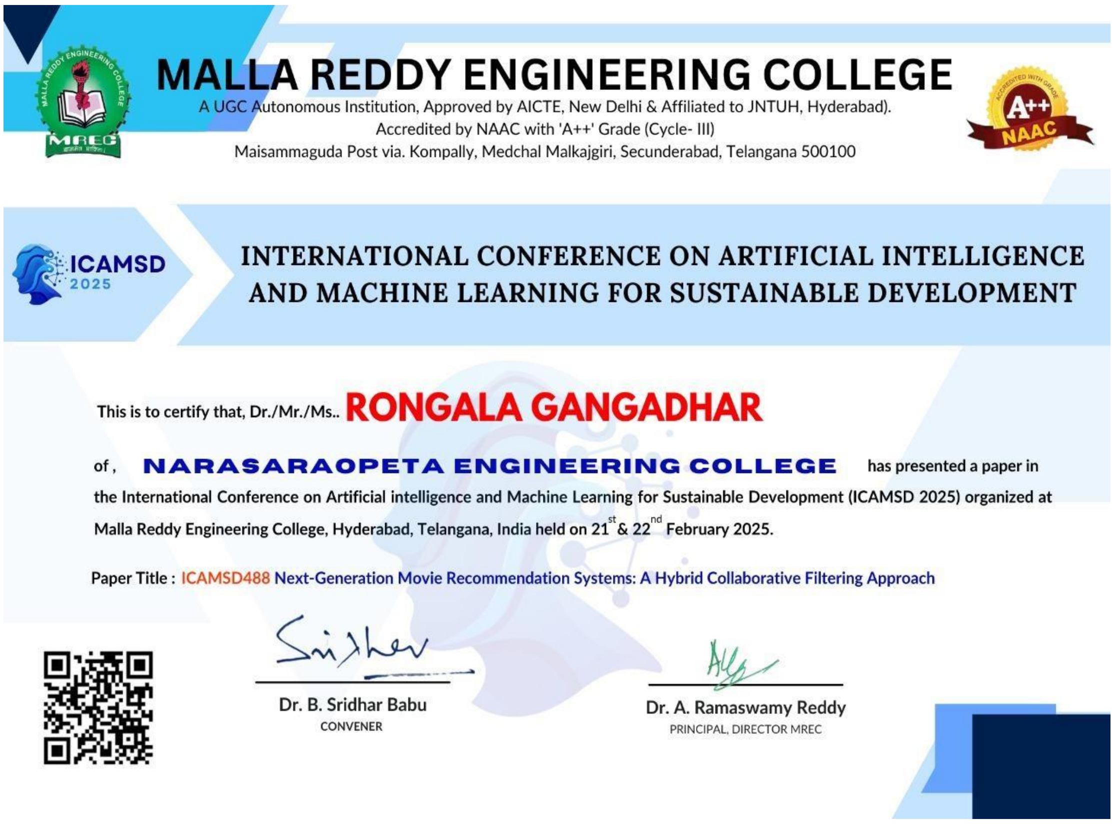
[illegible]

Dr. B. Sridhar Babu
CONVENER

[illegible]

Dr. A. Ramaswamy Reddy
PRINCIPAL, DIRECTOR MREC

Certificate – 3



MALLA REDDY ENGINEERING COLLEGE

A UGC Autonomous Institution, Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad).

Accredited by NAAC with 'A++' Grade (Cycle- III)

Maisammaguda Post via. Kompally, Medchal Malkajgiri, Secunderabad, Telangana 500100

ICAMSD 2025

INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING FOR SUSTAINABLE DEVELOPMENT

This is to certify that, Dr./Mr./Ms.. **RONGALA GANGADHAR**

of , **NARASARAOPETA ENGINEERING COLLEGE** has presented a paper in the International Conference on Artificial intelligence and Machine Learning for Sustainable Development (ICAMSD 2025) organized at Malla Reddy Engineering College, Hyderabad, Telangana, India held on 21st & 22nd February 2025.

Paper Title : ICAMSD488 Next-Generation Movie Recommendation Systems: A Hybrid Collaborative Filtering Approach

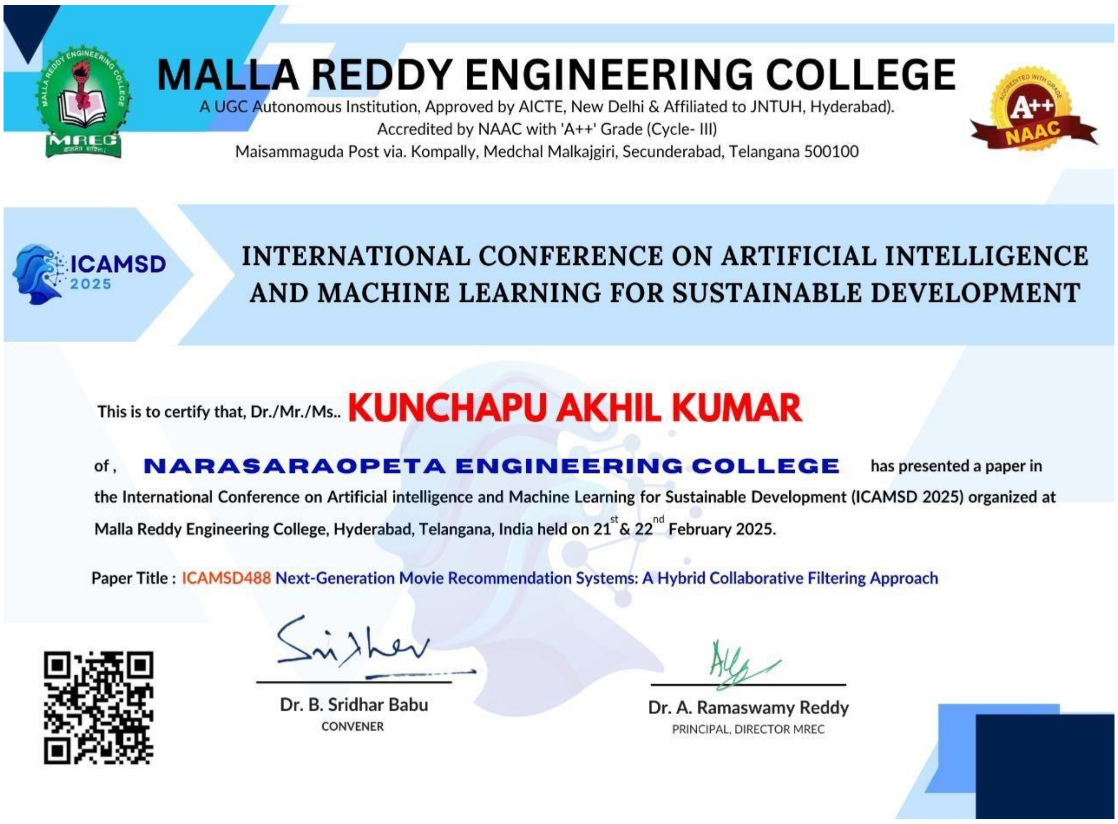
[illegible]

Dr. B. Sridhar Babu
CONVENER

[illegible]

Dr. A. Ramaswamy Reddy
PRINCIPAL, DIRECTOR MREC

Certificate - 4



MALLA REDDY ENGINEERING COLLEGE

A UGC Autonomous Institution, Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad).
Accredited by NAAC with 'A++' Grade (Cycle- III)
Maisammaguda Post via. Kompally, Medchal Malkajgiri, Secunderabad, Telangana 500100

ICAMSD 2025

INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING FOR SUSTAINABLE DEVELOPMENT

This is to certify that, Dr./Mr./Ms.. **KUNCHAPU AKHIL KUMAR**

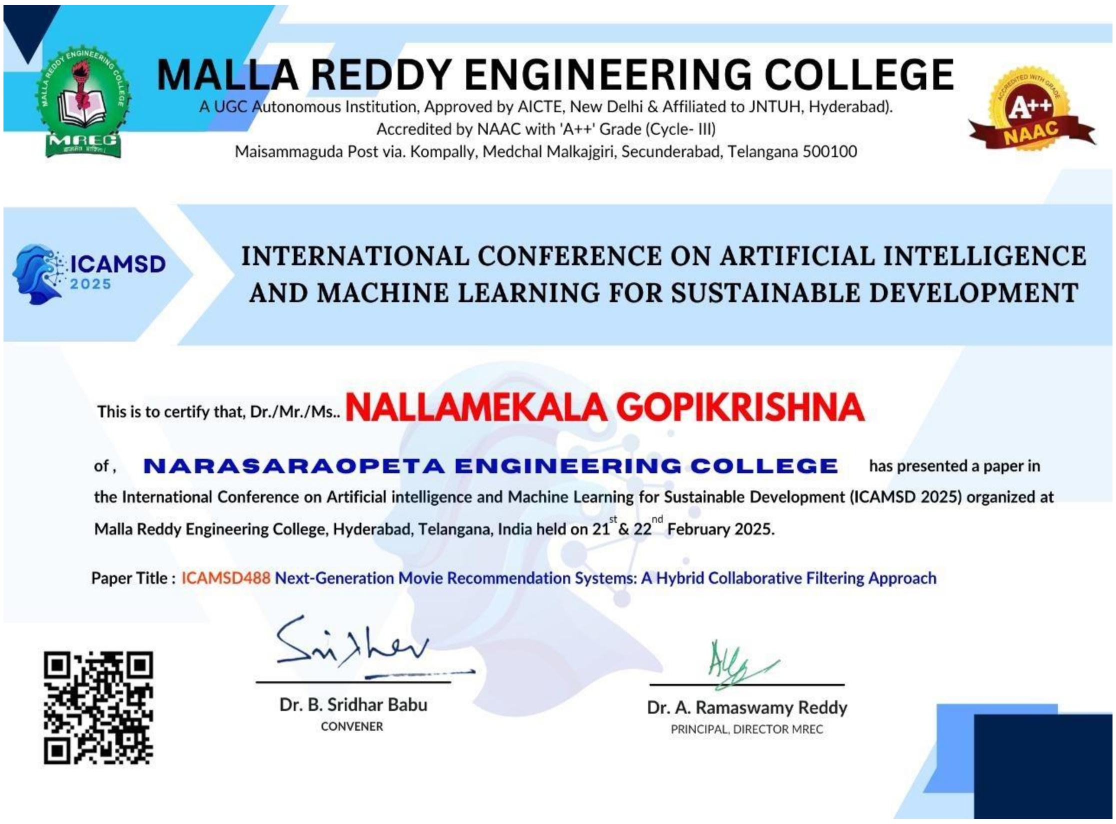
of , **NARASARAOPETA ENGINEERING COLLEGE** has presented a paper in the International Conference on Artificial intelligence and Machine Learning for Sustainable Development (ICAMSD 2025) organized at Malla Reddy Engineering College, Hyderabad, Telangana, India held on 21st & 22nd February 2025.

Paper Title : ICAMSD488 Next-Generation Movie Recommendation Systems: A Hybrid Collaborative Filtering Approach

Dr. B. Sridhar Babu
CONVENER

Dr. A. Ramaswamy Reddy
PRINCIPAL, DIRECTOR MREC

Certificate – 5



MALLA REDDY ENGINEERING COLLEGE

A UGC Autonomous Institution, Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad).
Accredited by NAAC with 'A++' Grade (Cycle- III)
Maisammaguda Post via. Kompally, Medchal Malkajgiri, Secunderabad, Telangana 500100

ICAMSD 2025

INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING FOR SUSTAINABLE DEVELOPMENT

This is to certify that, Dr./Mr./Ms.. NALLAMEKALA GOPIKRISHNA

of , NARASARAOPETA ENGINEERING COLLEGE has presented a paper in the International Conference on Artificial intelligence and Machine Learning for Sustainable Development (ICAMSD 2025) organized at Malla Reddy Engineering College, Hyderabad, Telangana, India held on 21st & 22nd February 2025.

Paper Title : ICAMSD488 Next-Generation Movie Recommendation Systems: A Hybrid Collaborative Filtering Approach

Dr. B. Sridhar Babu
CONVENER

Dr. A. Ramaswamy Reddy
PRINCIPAL, DIRECTOR MREC