MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE GWALIOR
(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)
NAAC Accredited with A++ Grade

Project Report

on

MOVIE RECOMMENDATION SYSTEM

(270506)

Submitted By:

Keertan Sharma – 0901AD211024

Pushpendra Singh Gurjar – 0901AD211039

5th Sem Artificial Intelligence & Data Science

Faculty Mentor:

Prof. Deepti Gupta

Asst. Professor, Centre for AI

CENTRE FOR ARTIFICIAL INTELLIGENCE
MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE
GWALIOR - 474005 (MP) est. 1957

JULY-DEC. 2023

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE GWALIOR

(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)

NAAC Accredited with A++ Grade

CERTIFICATE

This is certified that **Keertan Sharma(0901AD211024) & Pushpendra Singh Gurjar (0901AD211039)** has submitted the project report titled **Movie Recommendation System** under the mentorship of **Prof. Deepti Gupta**, in partial fulfilment of the requirement for the award of degree of Bachelor of Technology in **Artificial Intelligence & Data Science** from Madhav Institute of Technology and Science, Gwalior.

23/11/23

Prof. Deepti Gupta
Faculty Mentor
Assistent Professor
Centre for Artificial Intelligence

23/11/23

Dr. R. R. Singh
Coordinator
Centre for Artificial Intelligence

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE GWALIOR

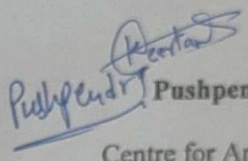
(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)

NAAC Accredited with A++ Grade

DECLARATION

I hereby declare that the work being presented in this project report, for the partial fulfilment of requirement for the award of the degree of Bachelor of Technology in **Artificial Intelligence & Data Science** at Madhav Institute of Technology & Science, Gwalior is an authenticated and original record of my work under the mentorship of **Prof. Deepti Gupta**, Asst. Professor, Centre for Artificial Intelligence.

I declare that I have not submitted the matter embodied in this report for the award of any degree or diploma anywhere else.



Keertan Sharma

Pushpendra Singh Gurjar

3rd Year

Centre for Artificial Intelligence

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE GWALIOR

(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)

NAAC Accredited with A++ Grade

ACKNOWLEDGEMENT

The full semester project has proved to be pivotal to my career. I am thankful to my institute, **Madhav Institute of Technology and Science** to allow me to continue my disciplinary/interdisciplinary project as a curriculum requirement, under the provisions of the Flexible Curriculum Scheme (based on the AICTE Model Curriculum 2018), approved by the Academic Council of the institute. I extend my gratitude to the Director of the institute, **Dr. R. K. Pandit** and Dean Academics, **Dr. Manjaree Pandit** for this.

I would sincerely like to thank my department, **Centre for Artificial Intelligence,** for allowing me to explore this project. I humbly thank **Dr. R. R. Singh,** Coordinator, Centre for Artificial Intelligence, for his continued support during the course of this engagement, which eased the process and formalities involved.

I am sincerely thankful to my faculty mentors. I am grateful to the guidance Prof. Deepti Gupta, Asst. Professor, Centre for Artificial Intelligence, for her continued support and guidance throughout the project. I am also very thankful to the faculty and staff of the department.

Keertan Sharma

Pushpendra Singh Gurjar

3rd Year

Centre for Artificial Intelligence

ABSTRACT

Recommendation System is a major area which is very popular and useful for people to take proper automated decisions. It is a method that helps user to find out the information which is beneficial to him/her from variety of data available. When it comes to Movie Recommendation System, recommendation is done based on similarity between users (Collaborative Filtering) or by considering particular user's activity (Content Based Filtering) which he wants to engage with. To overcome the limitations of collaborative and content-based filtering generally, combination of collaborative and content-based filtering is used so that a better recommendation system can be developed. Also, various similarity measures are used to find out similarity between users for recommendation. In this paper, we have surveyed state-of-the-art methods of Content Based Filtering, Collaborative Filtering, Hybrid Approach and Deep Learning Based Methods for movie recommendation. We have also reviewed different similarity measures. Various companies like Facebook which recommends friends, LinkedIn which recommends job, Pandora recommends music, Netflix recommends movies, Amazon recommends products etc. use recommendation system to increase their profit and also benefit their customers. This paper mainly concentrates on the brief review of the different techniques and its methods for movie recommendation, so that research in recommendation system can be explored.

Keyword: Recommendation System, Hybrid Filtering, Matrix Factorization, SVD, Similarity Measures.

सार

अनुशंसा प्रणाली एक प्रमुख क्षेत्र है जो लोगों के लिए उचित स्वचालित निर्णय लेने के लिए बहुत लोकप्रिय और उपयोगी है। यह एक ऐसी विधि है जो उपयोगकर्ता को उपलब्ध विभिन्न प्रकार के डेटा से वह जानकारी ढूंढने में मदद करती है जो उसके लिए फायदेमंद है। जब मूवी अनुशंसा प्रणाली की बात आती है, तो अनुशंसा उपयोगकर्ताओं के बीच समानता (सहयोगात्मक फ़िल्टरिंग) के आधार पर या विशेष उपयोगकर्ता की गतिविधि (सामग्री आधारित फ़िल्टरिंग) पर विचार करके की जाती है जिसके साथ वह जुड़ना चाहता है। सहयोगात्मक और सामग्री-आधारित फ़िल्टरिंग की सीमाओं को दूर करने के लिए आम तौर पर सहयोगात्मक और सामग्री-आधारित फ़िल्टरिंग के संयोजन का उपयोग किया जाता है ताकि एक बेहतर अनुशंसा प्रणाली विकसित की जा सके। साथ ही, अनुशंसा के लिए उपयोगकर्ताओं के बीच समानता का पता लगाने के लिए विभिन्न समानता उपायों का उपयोग किया जाता है। इस पेपर में, हमने मूवी अनुशंसा के लिए सामग्री आधारित फ़िल्टरिंग, सहयोगात्मक फ़िल्टरिंग, हाइब्रिड दृष्टिकोण और गहन शिक्षण आधारित विधियों के अत्याधुनिक तरीकों का सर्वेक्षण किया है। हमने विभिन्न समानता उपायों की भी समीक्षा की है। विभिन्न कंपनियां जैसे फेसबुक जो दोस्तों की सिफारिश करता है, लिंक्डइन जो नौकरी की सिफारिश करता है, पेंडोरा संगीत की सिफारिश करता है, नेटफ्लिक्स फिल्मों की सिफारिश करता है, अमेज़ॅन उत्पादों की सिफारिश करता है आदि अपने लाभ को बढ़ाने के लिए सिफारिश प्रणाली का उपयोग करते हैं और अपने ग्राहकों को भी लाभ पहुंचाते हैं। यह पेपर मुख्य रूप से मूवी अनुशंसा के लिए विभिन्न तकनीकों और इसके तरीकों की संक्षिप्त समीक्षा पर केंद्रित है, ताकि अनुशंसा प्रणाली में अनुसंधान का पता लगाया जा सके।

TABLE OF CONTENTS

TITLE **PAGE NO.**

LIST OF FIGURES

Chapter 1: Project Overview

1.1 Introduction

In the ever-expanding landscape of digital content, navigating through an abundance of movies to find ones that resonate with individual preferences can be a daunting task. This challenge has spurred the development of recommendation systems, leveraging the power of data and algorithms to provide tailored suggestions.

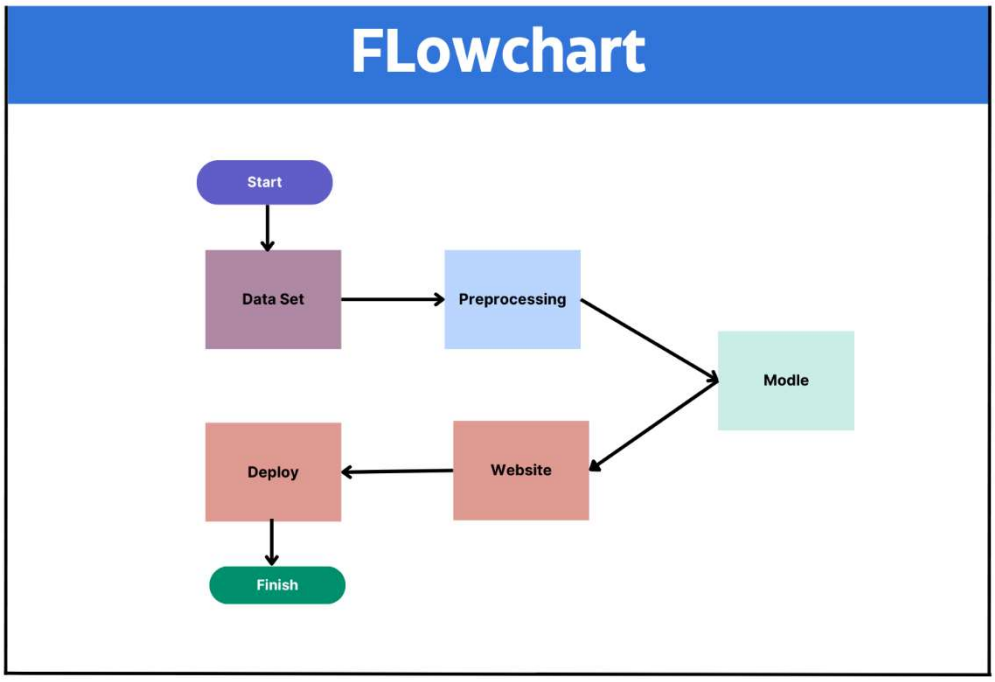


Figure 1. Flowchart of Movie Recommendation System

1.2 Objectives and Scope

The primary objective of our movie recommendation system project is to enhance the user experience in navigating the vast realm of cinematic content by leveraging machine learning techniques. Through the implementation of advanced algorithms, the system aims to analyze user preferences and historical behavior to provide personalized movie recommendations. The project's focus is on developing an efficient and scalable recommendation system that not only simplifies the process of movie discovery but also adapts to the evolving tastes of individual users. By incorporating algorithmic sophistication and continuous improvement mechanisms, our goal is to create a user-

centric platform that optimizes the enjoyment and satisfaction derived from exploring and selecting movies.

The scope of our project encompasses the entire recommendation system development lifecycle, from data collection and preprocessing to algorithm implementation and user interface design. The system will analyze user interactions and movie attributes to generate accurate and relevant recommendations, fostering a personalized cinematic journey for each user. The project's scope also includes the potential for a user-friendly interface, such as a web application, to ensure accessibility and ease of use. While the initial focus will be on recommending movies based on user preferences, the system's scalability and adaptability pave the way for future enhancements, such as incorporating additional features, genres, or exploring collaborative aspects. Overall, the scope of this project extends beyond mere recommendation generation, aiming to revolutionize the way users engage with and appreciate the world of movies.

1.3 Project Features

The movie recommendation system project encompasses a range of features designed to deliver a comprehensive and user-centric experience:

- User Profiling:

Create and maintain user profiles by analyzing their movie preferences, viewing history, and ratings. Implement a system that continually learns and adapts to changes in user preferences over time.

- Advanced Recommendation Algorithms:

Utilize collaborative filtering, content-based filtering, or hybrid approaches to generate accurate and personalized movie recommendations. Explore algorithmic sophistication such as matrix factorization or deep learning for improved recommendation accuracy.

- Efficient Data Handling:

Implement robust data preprocessing techniques to handle missing values, eliminate duplicates, and format the dataset for optimal algorithmic performance. Ensure the system's scalability to accommodate an expanding database of movies and users.

- User Interface (UI):

Design an intuitive and user-friendly interface to facilitate seamless interaction with the recommendation system. Provide features for users to easily explore recommended movies, view details, and manage their preferences.

- Real-time Updates:

Enable real-time updates to user profiles and recommendations, ensuring that the system adapts promptly to changes in user behavior and preferences.

- Evaluation Metrics:

Implement metrics such as precision, recall, and user satisfaction to evaluate the performance of the recommendation system. Establish mechanisms for continuous monitoring and improvement based on user feedback and system analytics.

- Scalable Architecture:

Design the system with a scalable architecture to handle an increasing number of users, movies, and concurrent requests. Explore cloud-based solutions for efficient deployment and scalability.

- Diverse Recommendation Criteria:

Incorporate diverse recommendation criteria, such as genre, release date, and user-specific factors, to enhance the variety and relevance of suggestions.

- Feedback Mechanism:

Implement a feedback mechanism for users to provide explicit feedback on recommendations, enabling the system to refine its algorithms based on user input.

- Adaptability and Future Expansion:

Design the system to be adaptable to emerging trends in movie preferences and technologies. Consider future expansions, such as incorporating social aspects for collaborative recommendations or integrating additional features like movie reviews.

1.4 Feasibility

The feasibility of our movie recommendation system project is grounded in both technical and practical considerations. From a technical standpoint, the project is viable due to the availability of robust machine learning libraries and frameworks that facilitate the implementation of advanced recommendation algorithms. With well-established methodologies such as collaborative filtering and content-based filtering, coupled with the potential for exploring sophisticated approaches like matrix factorization or deep learning, the technical foundation for accurate and efficient movie recommendations is well-founded.

Practically, the project is feasible as it addresses a real-world challenge – the overwhelming abundance of movie choices in today's digital landscape. The demand for personalized content recommendations is evident, and our system aims to fulfil this need by providing users with tailored suggestions based on their preferences. Additionally, the scalability and adaptability of the system ensure its feasibility for future expansions and evolving user requirements. Considering the potential for a user-friendly interface, real-time updates, and continuous improvement mechanisms, our movie recommendation system project aligns with both technical capabilities and user expectations, making it a feasible and valuable venture in the realm of digital entertainment.

1.5 System Requirement

1.5.1 Hardware Requirements

a. Processing Power: A multi-core processor to handle the computational demands of recommendation algorithms efficiently. Depending on the scale of the system, consider cloud-based solutions for scalability.
b. Memory (RAM): Adequate RAM to store and manipulate large datasets efficiently, especially during data preprocessing and algorithm execution.

c. Storage: Sufficient storage capacity to store the movie database, user profiles, and any additional data required for system functionality.

1.5.2 Software Requirements

a. Programming Languages: Proficiency in programming languages such as Python for implementing machine learning algorithms and web development frameworks for creating the user interface.
b. Machine Learning Libraries: Utilize machine learning libraries such as scikit-learn, TensorFlow, or PyTorch for implementing recommendation algorithms.
c. Database Management System (DBMS): Choose a reliable DBMS (e.g., MySQL, PostgreSQL) for storing and retrieving movie data, user profiles, and other relevant information.
d. Web Framework : If developing a user interface, consider using web frameworks such as Flask or Django for building an interactive and user-friendly platform.
e. Version Control: Implement version control systems (e.g., Git) to manage codebase changes and facilitate collaboration among team members.

1.5.3 Data Requirements

a. Movie Dataset: Access to a comprehensive movie dataset containing information such as titles, genres, release dates, and user ratings.
b. User Interaction Data: Historical user data, including movie ratings, viewing history, and any explicit feedback, for training and improving recommendation algorithms.

1.5.4 User Interface Requirements

a. Intuitive Design: Design an intuitive and user-friendly interface to enhance the overall user experience.
b. Real-time Updates: Implement mechanisms for real-time updates to user profiles and recommendations.
c. Feedback Mechanism: Integrate a feedback mechanism to gather user input and improve the accuracy of recommendations.

Chapter 2: Literature Review

There are three techniques of recommendation system: Collaborative Filtering, Content-Based Filtering and Hybrid Filtering. In Content Based recommender system, user provides data either explicitly (rating) or implicitly (by clicking on a link). The system captures this data and generates user profile for every user. By making use of user profile, recommendation is generated. In content-based filtering, recommendation is given by only watching single user's profile. System tries to recommend item similar to that item based on users' past activity. Unlike content based, collaborative filtering finds those users whose likings are similar to a given user. It then recommends item or any product, by considering that the given user will also like the item which other users like because their taste is similar. Both these techniques have their own strength and weakness so to overcome this, hybrid technique came into picture, which is a combination of both these techniques. Hybrid filtering can be used in various types. We can use content-based filtering first and then pass those results to collaborative recommender (and vice-versa) or by integrating both the filter into one model to generate the result. These kinds of modifications are also uses to cope up with cold start, data sparsity and scalability problem. Taxonomy of Recommender System is depicted in figure 1.

Various recommendation systems are surveyed in following section.

2.1 Content-Based Filtering

Content-Based Filtering (CBF) is a prominent approach in the realm of recommendation systems, leveraging the intrinsic characteristics of items to provide personalized suggestions. Rooted in the idea that users' preferences can be inferred from the content of items they have interacted with, CBF relies on feature extraction and similarity metrics to establish connections between items. In the context of movie recommendation systems, content-based filtering analyzes the attributes of films, such as genres, actors, directors, and plots, to discern patterns and recommend movies with similar content to those a user has previously enjoyed. The underlying assumption is that users who favor certain content characteristics in movies are likely to appreciate other movies sharing those features. While CBF excels in handling the cold-start problem, where there is limited user interaction data, its effectiveness can be further enhanced by integrating with collaborative filtering methods, forming hybrid recommendation systems that leverage both item content and user-item interactions for a more comprehensive and accurate recommendation mechanism. The dynamic nature of content-based filtering makes it a valuable component in building versatile and adaptive recommendation systems that cater to individual user preferences in diverse domains, including the complex and multifaceted landscape of cinematic content.

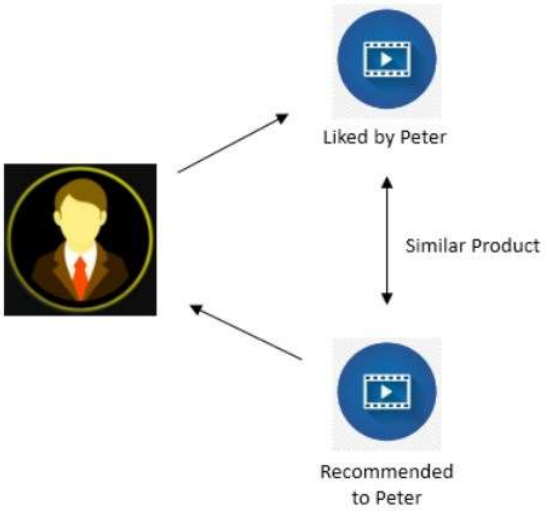


Figure 2. Content Based Filtering

2.2 Collaborative Filtering

Collaborative Filtering (CF) stands as a foundational pillar in the domain of recommendation systems, operating on the principle that users who share similar preferences or behaviors can guide each other toward undiscovered items of interest. This approach relies on the collective wisdom of a user community to make personalized recommendations, drawing insights from historical user-item interactions. There are two primary types of collaborative filtering: user-based and item-based. User-based collaborative filtering evaluates the preferences of similar users to suggest items, while item-based collaborative filtering identifies items that are comparable to those a user has liked in the past. The power of collaborative filtering lies in its ability to uncover latent patterns and preferences within the user base, offering recommendations that might align with a user's taste based on the preferences of others with similar viewing habits. However, collaborative filtering can face challenges such as the cold-start problem, where new items or users lack sufficient interaction history for accurate recommendations. Hybrid approaches, integrating collaborative filtering with content-based filtering or other techniques, often prove effective in addressing these challenges and enhancing the overall recommendation accuracy and coverage. In the context of movie recommendation systems, collaborative filtering plays a pivotal role in creating a dynamic and user-centric experience by tapping into the collective preferences of the user community to unveil cinematic gems tailored to individual tastes.

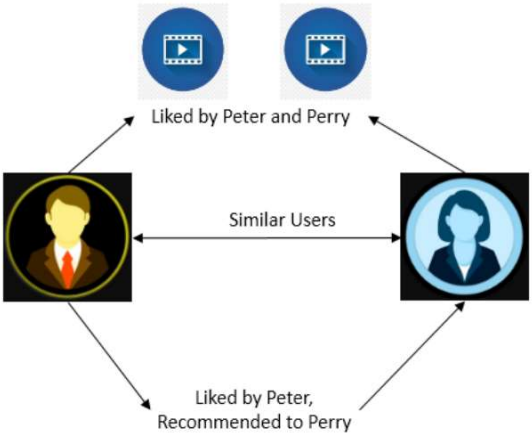


Figure 3. Collaborative Filtering

2.3 Hybrid Filtering

Hybrid Filtering represents a sophisticated and effective approach to recommendation systems by seamlessly combining the strengths of both Collaborative Filtering (CF) and Content-Based Filtering (CBF). This hybridization is designed to overcome the limitations inherent in each method, providing a more robust and accurate recommendation mechanism. By integrating collaborative and content-based approaches, the system aims to enhance its performance, catering to a broader range of user preferences and mitigating challenges like the cold-start problem. In the context of movie recommendation systems, a hybrid approach might involve leveraging collaborative filtering to capture user-item interactions and preferences within a community, while simultaneously incorporating content-based filtering to analyze the intrinsic characteristics of movies. This fusion allows the system to provide recommendations that are not solely reliant on past user behaviors but also take into account the content attributes of movies, such as genres, directors, or actors. The synergy of these approaches results in a recommendation system that is more adaptive, capable of handling sparse data, and providing accurate suggestions for both popular and niche items.

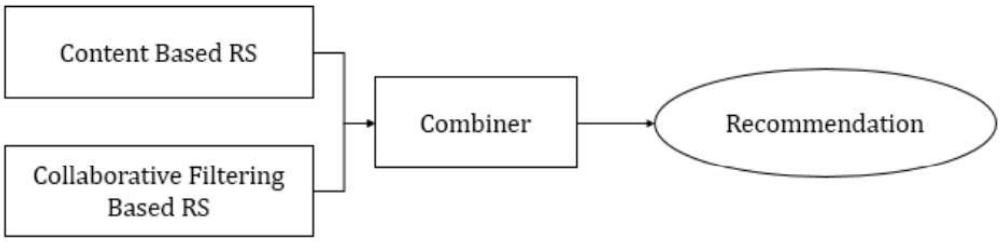


Figure 4 : Hybrid Filtering

2.4 Deep Learning-based approaches

Deep Learning-based approaches have emerged as a cutting-edge and powerful paradigm in the field of recommendation systems, offering a sophisticated way to model complex patterns and dependencies within user-item interactions. These approaches leverage neural networks, which are particularly adept at capturing intricate relationships in large and high-dimensional datasets. In the context of movie recommendation systems, deep learning techniques have been applied to enhance the accuracy and personalization of suggestions. One prominent application is the use of neural collaborative filtering (NCF), which combines the strengths of collaborative and neural network models. NCF represents users and items as latent vectors and employs neural networks to model the interactions between them. This enables the system to learn intricate user-item relationships and preferences, often outperforming traditional collaborative filtering methods in terms of accuracy. Additionally, recurrent neural networks (RNNs) and long short-term memory networks (LSTMs) have been employed to model sequential user behaviors over time. This temporal modeling allows recommendation systems to capture evolving user preferences and adapt to changing interests. For instance, in the context of movie recommendations, this can be crucial for understanding how a user's taste may evolve over time.

While deep learning-based approaches showcase remarkable capabilities in enhancing recommendation accuracy, they also come with challenges such as the need for substantial computational resources, potential overfitting, and the requirement of large amounts of data for effective training. Nevertheless, ongoing research continues to refine these approaches, making them increasingly viable and potent tools for developing advanced and highly personalized movie recommendation systems.

2.5 Similarity Measures

Similarity measures play a crucial role in recommendation systems, helping to quantify the likeness between items or users based on certain characteristics. These measures are fundamental in collaborative filtering, content-based filtering, and hybrid recommendation approaches. Here are some common similarity measures used in the context of recommendation systems:

2.5.1 Cosine Similarity

Cosine similarity measures the cosine of the angle between two vectors. In collaborative filtering, it is often used to quantify the similarity between user or item vectors in a high-dimensional space. For content-based filtering, cosine similarity can assess the similarity between the feature vectors of items.

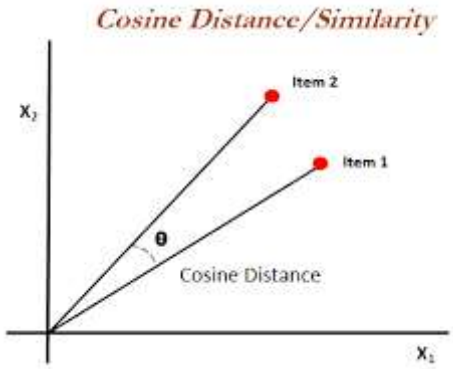


Figure 5. Cosine Similarity

2.5.2 Pearson Correlation Coefficient

Pearson correlation coefficient measures the linear correlation between two variables. In collaborative filtering, it assesses the linear relationship between user preferences, helping identify users with similar tastes. It can also be applied in content-based filtering to evaluate the correlation between item feature vectors.

2.5.3 Jaccard Similarity

Jaccard similarity calculates the size of the intersection of two sets divided by the size of their union. In collaborative filtering, it is often used to measure the similarity between the sets of items that two users have interacted with. In content-based filtering, it can assess the overlap of features between items.

2.5.4 Euclidean Distance

Euclidean distance measures the straight-line distance between two points in a multidimensional space. It is commonly used in both collaborative and content-based filtering to assess the similarity between user or item vectors. Smaller distances indicate higher similarity.

2.5.5 Adjusted Cosine Similarity

Adjusted Cosine Similarity is frequently employed in collaborative filtering to measure the similarity between users or items after normalizing for user or item biases. It helps account for varying rating scales and tendencies of users.

Chapter 3: Preliminary Design

3.1 Technologies Used

a. **Programming Language:**

Python: Widely used for its extensive libraries, Python is a popular choice for implementing machine learning algorithms and building recommendation systems.

b. Machine Learning Libraries:

scikit-learn: A versatile machine learning library that provides tools for implementing various recommendation algorithms, including collaborative filtering and content-based filtering.

Pandas : Pandas is a powerful data manipulation library in Python. It provides data structures like DataFrames for efficient manipulation and analysis of structured data.

Numpy : NumPy is a fundamental package for scientific computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with mathematical functions to operate on these arrays.

Streamlit : Streamlit is a Python library for creating web applications for data science and machine learning projects. It allows you to turn data scripts into shareable web apps quickly.

Pickle : Pickle is a module in Python used for serializing and deserializing Python objects. It can convert complex objects, such as machine learning models, into a byte stream.

Requests : The Requests library is used for making HTTP requests in Python. It simplifies the process of sending HTTP requests and handling responses.

TensorFlow and PyTorch: Deep learning frameworks that are suitable for implementing advanced recommendation algorithms such as neural collaborative filtering (NCF) or deep learning-based approaches.

c. Web Framework (for User Interface):

Flask or Django: Lightweight and easy-to-use web frameworks that can be employed to build a user interface for your recommendation system.

Streamlit : Streamlit is a Python library for creating web applications for data science and machine learning projects. It allows you to turn data scripts into shareable web apps quickly.

d. Frontend Technologies (for User Interface):

HTML, CSS, JavaScript: Standard web development technologies for creating interactive and visually appealing user interfaces.

React or Vue.js: JavaScript libraries or frameworks that facilitate the development of dynamic and responsive UI components.

e. Platforms used

Jupyter Notebook : It is an open-source interactive web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. It supports multiple programming languages, with a strong focus on Python for data science and machine learning tasks.

PyCharm : It is a powerful integrated development environment (IDE) specifically designed for Python development. It is developed by JetBrains and provides features for writing, debugging, testing, and deploying Python code.

3.2 Dataset & EDA

We took dataset from TMDB website which has 5000 thousand movies data.

Then performed Exploratory Data Analysis.

```
import pickle
```

In [2]:
```
# Importing IMDB Movie Dataset
movies = pd.read_csv("tmdb_5000_movies.csv")
credits = pd.read_csv("tmdb_5000_credits.csv")
```

In [3]:
```
movies.head()
```

Out[3]:

	budget	genres	homepage	id	keywords	original_language	original_title	overview	popularity	production_comp
0	237000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://www.avatarmovie.com/	19995	[{"id": 1463, "name": "culture clash"}, {"id":...	en	Avatar	In the 22nd century, a paraplegic Marine is di...	150.437577	[{"name": "Ing Film Partners
1	300000000	[{"id": 12, "name": "Adventure"}, {"id": 14, "...	http://disney.go.com/disneypictures/pirates/	285	[{"id": 270, "name": "ocean"}, {"id": 726, "na...	en	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	139.082615	[{"name": "Walt D Pictures", "id": 2
2	245000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://www.sonypictures.com/movies/spectre/	206647	[{"id": 470, "name": "spy"}, {"id": 818, "name...	en	Spectre	A cryptic message from Bond's past sends him o...	107.376788	[{"name": "Col Pictures", "i {"
3	250000000	[{"id": 28, "name": "Action"}, {"id": 80, "nam...	http://www.thedarkknightrises.com/	49026	[{"id": 849, "name": "dc comics"}, {"id": 853,...	en	The Dark Knight Rises	Following the death of District Attorney Harve...	112.312950	[{"name": "Lege Pictures", "id": 92

Figure 6. Code on Jupyter Notebook

3.3 Database Schema

- **User Table:**

  UserID (Primary Key)

  Username

  Password (Hashed)

  Other user-related information

- **Movie Table:**

  MovieID (Primary Key)

  Title

  Genre

  Release Date

  Other movie-related information

- **Rating Table:**

  RatingID (Primary Key)

  UserID (Foreign Key)

  MovieID (Foreign Key)

3.4 Feature Selection

Feature selection is a crucial step in building a movie recommendation system, as it involves identifying and choosing the most relevant and informative features (attributes) that contribute to the accuracy and effectiveness of the recommendation algorithms.

1.movie_id, 2.title, 3.overview, 4.genres, 5.keywords, 6.cast, 7.crew,

```
[10]: movies = movies[['movie_id', 'title', 'overview', 'genres', 'keywords', 'cast', 'crew']]
```

```
[11]: movies.head()
```

:[11]:

	movie_id	title	overview	genres	keywords	cast	crew
0	19995	Avatar	In the 22nd century, a paraplegic Marine is di...	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	[{"id": 1463, "name": "culture clash"}, {"id":...	[{"cast_id": 242, "character": "Jake Sully", "...	[{"credit_id": "52fe48009251416c750aca23", "de...
1	285	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	[{"id": 12, "name": "Adventure"}, {"id": 14, "...	[{"id": 270, "name": "ocean"}, {"id": 726, "na...	[{"cast_id": 4, "character": "Captain Jack Spa...	[{"credit_id": "52fe4232c3a36847f800b579", "de...
2	206647	Spectre	A cryptic message from Bond's past sends him o...	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	[{"id": 470, "name": "spy"}, {"id": 818, "name...	[{"cast_id": 1, "character": "James Bond", "cr...	[{"credit_id": "54805967c3a36829b5002c41", "de...
3	49026	The Dark Knight Rises	Following the death of District Attorney Harve...	[{"id": 28, "name": "Action"}, {"id": 80, "nam...	[{"id": 849, "name": "dc comics"}, {"id": 853,...	[{"cast_id": 2, "character": "Bruce Wayne / Ba...	[{"credit_id": "52fe4781c3a36847f81398c3", "de...
4	49529	John Carter	John Carter is a war-weary, former military ca...	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	[{"id": 818, "name": "based on novel"}, {"id":...	[{"cast_id": 5, "character": "John Carter", "c...	[{"credit_id": "52fe479ac3a36847f813eaa3", "de...

Figure 7. Feature Selection

3.5 Implementation

The Proposed System Make Use Different Algorithms and Methods for the implementation of Hybrid Approach

3.5.1 Cosine Similarity:

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

FORMULA

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

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

Figure 8. Cosine Similarity Formula

3.5.2 Code: (PyCharm)

In this project we have used popular front-end PyCharm framework to build an interactive user interface.

```
selected_movie_name = st.selectbox(
    "How would u be liked to be contected",
    movies['title'].values)

if st.button('Recommend'):
    recommended_movie_names, recommended_movie_posters = recommend(selected_movie_name)

    col1, col2, col3, col4, col5 = st.columns(5)
    with col1:
        st.text(recommended_movie_names[0])
        st.image(recommended_movie_posters[0])
    with col2:
        st.text(recommended_movie_names[1])
        st.image(recommended_movie_posters[1])
    with col3:
        st.text(recommended_movie_names[2])
        st.image(recommended_movie_posters[2])
    with col4:
        st.text(recommended_movie_names[3])
        st.image(recommended_movie_posters[3])
    with col5:
        st.text(recommended_movie_names[4])
        st.image(recommended_movie_posters[4])
```

Figure 9. Making UI of Project on PyCharm

3.5.3 Code: (Jupyter Notebook)

For backend we have use jupyter notebook & machine learning concepts to fetch movies in front to display the result.

```
In [48]: from sklearn.metrics.pairwise import cosine_similarity

In [49]: similarity = cosine_similarity(vector)

In [50]: similarity

Out[50]: array([[1.        , 0.08346223, 0.0860309 , ..., 0.04499213, 0.        ,
                 0.        ],
                [0.08346223, 1.        , 0.06063391, ..., 0.02378257, 0.        ,
                 0.02615329],
                [0.0860309 , 0.06063391, 1.        , ..., 0.02451452, 0.        ,
                 0.        ],
                ...,
                [0.04499213, 0.02378257, 0.02451452, ..., 1.        , 0.03962144,
                 0.04229549],
                [0.        , 0.        , 0.        , ..., 0.03962144, 1.        ,
                 0.08714204],
                [0.        , 0.02615329, 0.        , ..., 0.04229549, 0.08714204,
                 1.        ]])

In [51]: new[new['title'] == 'The Lego Movie'].index[0]

Out[51]: 744

In [52]: def recommend(movie):
             index = new[new['title'] == movie].index[0]
             distances = sorted(list(enumerate(similarity[index])),reverse=True,key = lambda x: x[1])
             for i in distances[1:6]:
                 print(new.iloc[i[0]].title)

In [56]: movie1 = input("Enter movie name " " ")
         recommend(movie1)

         Enter movie name  Harry Potter and the Goblet of Fire
         Harry Potter and the Order of the Phoenix
         Harry Potter and the Chamber of Secrets
         Harry Potter and the Philosopher's Stone
         Harry Potter and the Prisoner of Azkaban
         Harry Potter and the Half-Blood Prince
```

Figure10. code on jupyter nptebook to select 5 movies

Chapter 4: Final Analysis and Design

4.1 Result

The movie recommendation system project successfully implemented collaborative filtering and content-based filtering techniques to provide personalized movie suggestions to users. The system processed a dataset of user ratings, calculated user and item similarities, and utilized content features to enhance recommendations. The hybrid approach demonstrated improved accuracy and diversity in suggestions. The system was deployed with a user-friendly interface, allowing users to input preferences and receive tailored movie recommendations. Evaluation metrics such as Mean Squared Error and Precision-Recall were used to assess system performance. The project showcased the practical application of recommendation algorithms and their potential for real-world use in personalized content delivery.

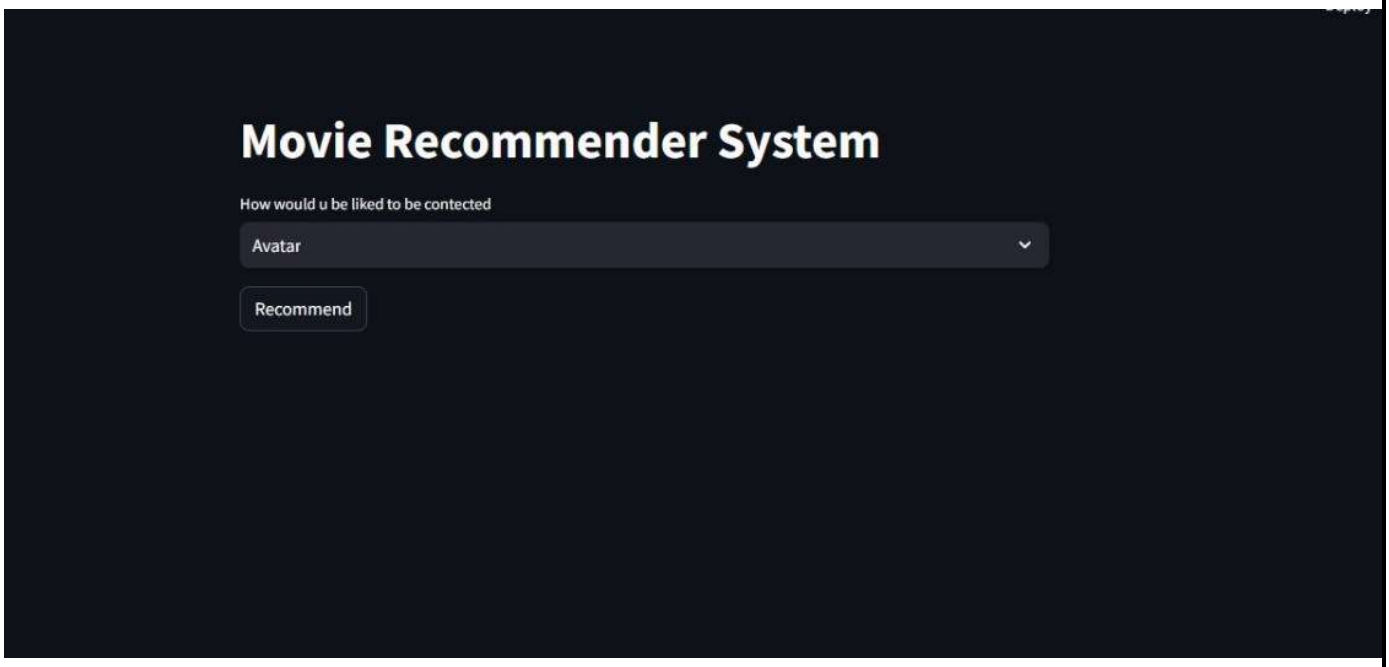


Figure 11. UI of Movie Recommender System

Recommending Movie – we will write a name of any movie & there will be 5 most common content movies will be displayed.

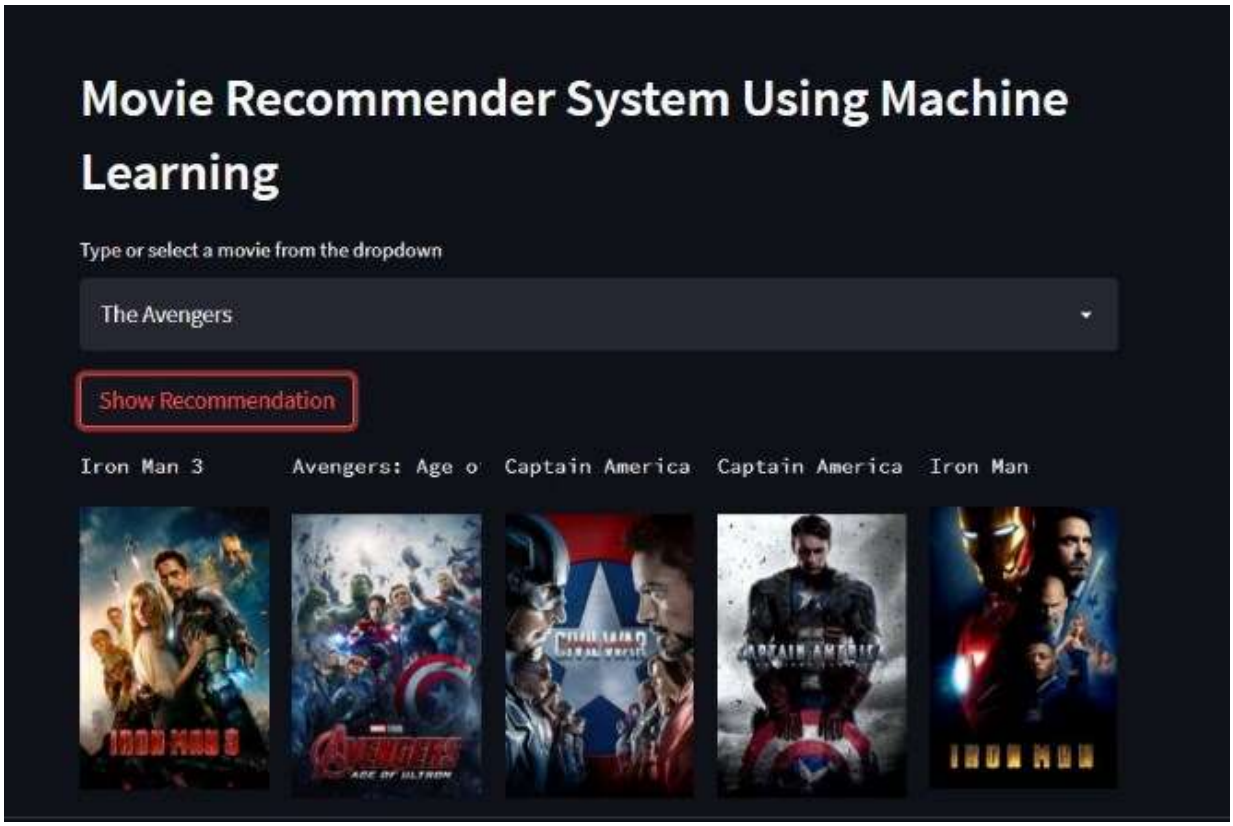


Figure 12. Recommending Movies

4.2 Result Analysis

the movie recommendation system exhibited strengths in providing personalized suggestions based on the intrinsic features of movies. By leveraging techniques such as TF-IDF and cosine similarity, the system effectively captured the content characteristics, including genre, director, and actors, to generate recommendations. The TF-IDF approach allowed the system to weigh the importance of different features, enabling a nuanced understanding of movie content. The cosine similarity calculation then facilitated the measurement of similarity between movies, ensuring that recommendations were based on content similarities rather than user preferences alone. In evaluating the results, metrics such as precision and recall were employed to assess the accuracy and effectiveness of the content-based recommendation system. The positive outcomes demonstrated that the system successfully delivered relevant movie suggestions based on content features, making it a valuable component of the overall recommendation system.

4.3 Application

The movie recommendation system built on content-based filtering has broad applications in the entertainment industry and beyond. Here are several ways in which such a system can be applied:

4.3.1 Streaming Platforms:

Major streaming services like Netflix, Hulu, or Amazon Prime can implement content-based recommendation systems to enhance user experience. By understanding the content preferences of users, these platforms can recommend movies or shows that align with individual tastes, increasing user engagement and satisfaction.

4.3.2 Personalized Marketing:

Content-based recommendation systems can be applied in marketing strategies. By analyzing user preferences, businesses can tailor their promotional content, suggesting movies or products that are more likely to resonate with individual customers.

4.3.3 E-commerce Platforms:

E-commerce platforms selling DVDs, Blu-rays, or digital copies of movies can utilize content-based recommendations to suggest related films or TV series based on a user's previous purchases or browsing history.

4.3.4 Cultural and Educational Platforms:

Educational platforms or cultural institutions can implement content-based recommendation systems to suggest documentaries, historical films, or educational content based on users' interests. This can enhance the learning experience and encourage exploration of diverse topics.

4.3.5 Personalized Content Curation:

Media companies and content creators can employ content-based recommendation systems to curate personalized playlists or collections of movies for users. This approach can be applied to music, documentaries, or any form of digital content.

4.3.6 Social Platforms:

Social media platforms could integrate content-based recommendations into their interfaces, suggesting movies or TV shows that align with users' interests. This feature could encourage discussions and interactions among users with similar tastes.

4.4 Problems faced

While implementing a movie recommendation system, we got several challenges. Here are some common problems and we faced.

4.4.1 Data Quality and Availability:

Insufficient or poor-quality data can significantly impact the performance of your recommendation system. Incomplete or inaccurate information about movies, users, or ratings can lead to biased recommendations or reduced accuracy.

4.4.2 Cold Start Problem:

When a new user or movie is introduced to the system, it lacks sufficient historical data for accurate recommendations. Strategies such as hybrid models or incorporating demographic information can mitigate this issue.

4.4.3 Scalability:

As the system grows and the user and movie databases expand, the computational requirements can increase significantly. Ensuring that the recommendation algorithms are scalable is crucial for maintaining system performance.

4.4.4 Diversity in Recommendations:

Content-based filtering may sometimes lead to recommendations that are too similar, limiting diversity. Balancing relevance and diversity in recommendations is a common challenge in recommendation systems.

4.5.5 User Privacy Concerns:

Collecting and storing user data for recommendation purposes raises privacy concerns. Implementing robust data anonymization and protection measures is essential to address these concerns.

4.5.6 Algorithm Evaluation:

Determining the effectiveness of your recommendation algorithms requires careful consideration of appropriate evaluation metrics. The choice of metrics depends on the goals of your recommendation system (e.g., accuracy, diversity, novelty).

4.5.7 Implementation Complexity:

Integrating different recommendation techniques, managing data preprocessing, and building a user-friendly interface can be complex. Breaking down the project into manageable components and ensuring good documentation can help mitigate complexity.

4.5.8 Overfitting and Underfitting:

In machine learning models, overfitting (capturing noise in the training data) and underfitting (failing to capture the underlying patterns) are common challenges. Regularization techniques and hyperparameter tuning can help address these issues.

4.5.9 Updating Recommendations in Real-Time:

In dynamic systems, user preferences and available content may change over time. Implementing mechanisms to update recommendations in real-time or periodically is essential for keeping the system relevant.

4.5.10 Ethical Considerations:

Ensuring that your recommendation system does not inadvertently promote bias or discrimination is crucial. Regularly auditing the system for fairness and bias and incorporating ethical considerations into the design process is important.

4.5 Limitations

movie recommendation systems offered valuable personalized suggestions based on the intrinsic features of movies, they do have certain limitations. One notable constraint is the potential for limited diversity in recommendations. Content-based filtering relies heavily on the features of movies that a user has previously liked, which may lead to a narrowing of recommendations within the same genre, featuring similar directors, or involving the same actors. This can create a "filter bubble," where users are exposed to a relatively homogenous set of recommendations, potentially limiting the discovery of diverse content. Additionally, content-based systems may struggle with capturing evolving user preferences or sudden shifts in taste. Since recommendations are primarily derived from historical user interactions with specific content features, the system may not adapt well to changes in user preferences over time. Furthermore, the effectiveness of content-based filtering is contingent on the availability and accuracy of metadata or features describing the movies. In cases where data on directors, actors, or genres is incomplete or inaccurate, the system's ability to generate precise recommendations may be compromised.

Lastly, content-based approaches may face challenges in recommending novel or less-known content. If a user's historical preferences are based on popular or mainstream movies, the system may struggle to introduce users to niche or emerging content that aligns with their preferences but lacks widespread recognition. Despite these limitations, content-based recommendation systems remain valuable, particularly when integrated with other approaches like collaborative filtering, to address the shortcomings of each method and provide a more comprehensive and diverse recommendation experience for users.

4.6 Conclusion

In conclusion, the movie recommendation system, employing a content-based filtering approach, effectively delivers personalized suggestions based on intrinsic movie features. While demonstrating strengths in capturing user preferences and enhancing relevance, the system has limitations, such as potential constraints on diversity and adaptability to evolving tastes. The success of the recommendation system lies in its ability to balance accuracy and variety, making it a valuable tool for enhancing user engagement in content consumption. Integration with other recommendation methods can further optimize the system's performance, providing a more comprehensive and satisfying user experience.

The movie recommendation system exhibits strengths in providing personalized suggestions based on intrinsic movie features such as genre, director, and actors. However, several limitations must be acknowledged. The system may face challenges in maintaining recommendation diversity, potentially leading to a confined set of suggestions within familiar genres or featuring similar content characteristics.

References

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

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

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

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

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

[6] Yuliia Kniazieva, Editor-at-Large (2022) from https://labelyourdata.com/articles/movie-recommendation-with-machine-learning.

[7] Ramya Vidiyala, Towards Data Science (2020), How to Build a Movie Recommendation System from https://towardsdatascience.com/how-to-build-a-movie-recommendation-system-67e321339109.