Movie Recommendation System

Minor Project – I

Submitted By:

Sanskar Khandelwal	**21103172**	**B6**
Priyanshu Jain	**21103205**	**B8**
Yash Aeron	**21103207**	**B8**

Under the Supervision of :

Ms.Kirti Jain

December 2023

Submitted in partial fulfilment of the Degree of Bachelor of Technology

In

Computer Science Engineering

DEPARTMENT OF COMPUTER SCIENCE TECHNOLOGY & INFORMATION TECHNOLOGY

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA

TABLE OF CONTENTS

DECLARATION

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

Place: JIIT-62
Date: 1 Dec,2023

Name: Sanskar Khandelwal
Enrollment No. 21103172

Name: Yash Aeron
Enrollment No. 21103207

Name: Priyanshu Jain
Enrollment No. 21103205

CERTIFICATE

This is to certify that the work titled " **Movie Recommendation System**" submitted by **Sanskar Khandelwal, Yash Aeron and Priyanshu Jain** of BTech of Jaypee Institute of Information Technology, Noida has been carried out under our supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of any other degree or diploma.

Name of Supervisors:
Ms.Kirti Jain
Signature of Supervisors :

Designation:
Assistant Professor(Grade II)
Date:1 Dec,2023

ACKNOWLEDGEMENT

We would like to place our deep sense of gratitude to Ms.Kirti Jain of Jaypee Institute of Information Technology, Noida for their generous guidance help, and useful suggestions

We also wish to extend our thanks to other classmates for their insightful comments and constructive suggestions to improve the quality of this project work.

Sanskar Khandelwal (21103172)
Priyanshu Jain (21103205)
Yash Aeron (21103207)

Summary

The content-based movie recommendation system employs a sophisticated approach to enhance user experience by offering personalized movie suggestions based on the inherent content features of films. Beginning with the loading and merging of two datasets containing movie and credit information, the system undertakes a thorough data cleaning and transformation process. This involves extracting essential details such as genres, keywords, cast, and crew information from JSON-like string columns.

To facilitate numerical analysis, the 'tags' column undergoes text vectorization using the CountVectorizer from the scikit-learn library. This process converts the textual data into a numerical format, forming a matrix of token counts. The next crucial step involves calculating cosine similarity between the vectors representing each movie. Cosine similarity is a mathematical measure that quantifies the similarity of two vectors, allowing the system to determine how closely related movies are based on their content.

The recommendation function takes user input in the form of a selected movie title, identifies its index in the dataset, and then suggests five other movies with the highest cosine similarity scores. This personalized recommendation approach is grounded in the idea that movies with similar content features are likely to appeal to the same audience. By emphasizing intrinsic movie characteristics such as themes, genres, and key personnel, the content-based system caters to individual user preferences, delivering movie suggestions that align with their tastes.

Ultimately, this content-based recommendation system contributes to a more tailored and enjoyable movie-watching experience. Its reliance on content features, rather than collaborative filtering or user preferences, offers a unique approach to movie recommendations that can be particularly beneficial for users seeking suggestions based on the intrinsic qualities of films.

Signature of student Signature of Supervisor

Name: Sanskar Khandelwal

Name: Priyanshu Jain

Name: Yash Aeron

Date: 1 Dec,2023

Chapter 1

INTRODUCTION

1.1 General Introduction

In an era marked by an overwhelming abundance of cinematic choices, the development of a Movie Recommendation System using Machine Learning (ML) emerges as a pivotal response to the challenge of content navigation and discovery. The sheer volume of available movies, spanning diverse genres, directors, and themes, poses a significant dilemma for users seeking personalized and relevant viewing options. Traditional approaches to content recommendation often fall short in capturing the nuanced and evolving preferences of individual users. In response to this, the Movie Recommendation System harnesses the power of ML to revolutionize the way audiences engage with films. By delving into intricate patterns within extensive datasets, including user viewing history, ratings, and demographic information, the system aims to decode the complex tapestry of user preferences. Through a strategic combination of collaborative filtering, content-based filtering, and advanced ML algorithms such as matrix factorization and neural networks, the system not only offers tailored movie suggestions but also adapts in real-time to changing user tastes and emerging cinematic trends. The ultimate goal is to provide users with a curated and dynamic cinematic experience, fostering increased satisfaction and engagement in the ever-expanding landscape of digital entertainment.

A Recommender System is a system that learns the attributes of users from a given dataset provided and predicts items and then recommends them the top items from the system generated list. So we can say that the recommender system is personalized to particular users to recommend them.

1.2 Problem Statement:

The modern era of digital streaming platforms offers an extensive collection of movies and TV shows across various genres, making it challenging for users to make informed choices. The problem is twofold:

a. users need assistance in finding content that aligns with their preferences

b. streaming platforms seek to engage users by keeping them active

However, many individuals struggle to discover and select movies or TV shows to watch from the vast and ever-growing catalog of content available on streaming platforms. This abundance of choices often leads to decision paralysis, frustration, and a less satisfying user experience.

1.3 Significance of the Problem

The significance of the content discovery challenge in the digital streaming era is paramount as it directly influences the user experience and the sustainability of streaming platforms. In an age where an abundance of diverse content is readily available, the twofold nature of the problem underscores its impact on users' ability to make meaningful choices aligned with their preferences and on streaming platforms' imperative to keep users actively engaged. The persistent struggle individuals face in navigating this extensive catalog results in decision paralysis, leading to frustration and ultimately diminishing the overall satisfaction of the user experience. This challenge not only threatens user retention and the competitive edge of streaming platforms but also underscores the critical need for innovative solutions that enhance content discovery, ensuring that users can effortlessly connect with content tailored to their tastes, thereby shaping the success and longevity of digital streaming services in the evolving landscape of entertainment consumption.

1.4 Empirical Study:

In this empirical study, we rigorously assess the efficacy of a content-based movie recommendation system, focusing on leveraging content attributes such as genre, keywords, and user preferences. The research involves comprehensive data collection from diverse movie datasets, the implementation of a content-based recommendation algorithm, and meticulous evaluation using precision, recall, and F1-score metrics. Our analysis includes user feedback through surveys and interviews, aiming to gauge the system's accuracy and user satisfaction. Through this study, we seek to contribute valuable insights into the strengths, limitations, and potential enhancements of content-based movie recommendation systems, shedding light on their effectiveness in providing personalized and relevant movie suggestions to users.

1.5 Solution Approach:

Our solution involves creating a movie recommendation system that leverages machine learning and data analysis techniques to deliver tailored content recommendations to users. The system will consider a range of factors, including users' viewing history, ratings, genre preferences, and demographic information, as well as the intrinsic characteristics of the content itself, such as genre, director, actors, release year, and user-generated tags.

Key Objectives:

Personalization: The recommendation system will aim to provide personalized movie and TV show suggestions to each user, based on their unique viewing history and preferences. Personalization enhances user satisfaction and engagement.

Content Discovery: The system will help users discover new movies and TV shows that match their interests, thereby addressing the issue of choice overload and aiding users in finding hidden gems.

Recommendation Accuracy: Implement sophisticated recommendation algorithms that accurately predict user preferences and continually improve recommendation quality over time.

User Engagement: Increase user engagement on the platform by assisting users in selecting content they are likely to enjoy. This can lead to longer viewing sessions, higher user retention, and increased platform loyalty.

1.6 Comparison of existing approaches to the problem:

Movie recommendation systems employ collaborative filtering, relying on user behavior data to recommend items based on similar user preferences. User-based collaborative filtering suggests items liked by users with comparable tastes, while item-based collaborative filtering recommends items similar to those previously interacted with by a user. However, collaborative filtering faces challenges like the cold start problem for new users or items with limited historical data.

In contrast, content-based filtering recommends items based on intrinsic features such as genres, keywords, and cast information, making it less susceptible to the cold start problem. Yet, it may lack diversity, suggesting items with similar characteristics.

Hybrid approaches blend collaborative and content-based methods to offer accurate and diverse recommendations. These models can leverage collaborative filtering for initial suggestions and refine them using content-based filtering.

Matrix factorization techniques, including SVD and ALS, decompose the user-item interaction matrix to unveil latent features. While effective for handling sparse data, these methods may face scalability and real-time update challenges.

Deep learning models like neural collaborative filtering excel in capturing complex relationships but require substantial computational resources and extensive datasets.

Chapter 2:

2. LITERATURE SURVEY

2.1 Summary of paper studied:

Year	Title of the paper	Name of the Algorithm	Pros	Cons
2018	Content-Based Movie Recommendation System Using Genre Correlation	Content-Based Filtering.	Content-based Systems offer personalized recommendations based on user preferences, enhancing user satisfaction.	Content-based systems may lead to recommendations with limited diversity, as they tend to suggest items with similar features.

2018	A Content-Based Recommender System for Computer Science Publications	Content based filtering approach	The development of a content-based recommender system for computer science and technology, coupled with the creation of a dataset through web crawling, represents an innovative approach with potential applications in e-libraries and e-business.	While achieving a 61.37% accuracy in paper recommendation, the system acknowledges room for improvement, indicating potential challenges in achieving higher accuracy and F-measure, which will be addressed in future work

2018	Movie Recommendation System Using Collaborative Filtering.	Collaborative Filtering	The implemented movie recommendation system using collaborative filtering leverages Apache Mahout to provide effective movie suggestions by considering user ratings, enhancing user experience.	The current system relies primarily on user ratings, potentially limiting the diversity and accuracy of recommendations by not incorporating additional features like genre, directors, and actors.

2.2 Integrated summary of the literature studied:

The project entails the creation of a content-based movie recommendation system using genre correlation, employing a Movie Lens dataset with genre categorizations and user ratings. The objective is to furnish users with personalized movie suggestions based on their genre preferences and prior rating patterns. The methodology involves constructing a genre matrix and converting user ratings into a binary format for simplicity. The dot product of these matrices is computed, resulting in a binary format result matrix. By calculating Euclidean distances between users, the system identifies movies with the least deviation from a user's profile, offering tailored recommendations.

Simulation results illustrate the systematic process, from constructing the genre matrix to the final recommendation output based on Euclidean distances. The report concludes that the system effectively delivers personalized movie recommendations, showcasing the potential of content-based filtering with genre correlation.

Future work is proposed to explore advanced recommendation algorithms, integrate user feedback mechanisms, assess scalability, incorporate mobile cloud computing for energy efficiency, address security concerns, and evaluate system performance using relevant metrics. These enhancements aim to optimize the recommendation system and align it with evolving trends in movie recommendations and mobile cloud computing, ensuring ongoing relevance and efficiency.

Chapter 3

REQUIREMENT ANALYSIS AND SOLUTION APPROACH

3.1 Over all description of the project:

The project aims to implement a content-based movie recommendation system using Python and various data processing and natural language processing (NLP) techniques. The core of the project involves leveraging movie data from two datasets, 'tmdb_5000_movies.csv' and 'tmdb_5000_credits.csv.' These datasets are loaded into Pandas DataFrames, merged, and cleaned to create a consolidated dataset with essential movie details.

The data is then transformed using custom functions to extract relevant information from JSON-like string columns, such as genres, keywords, cast, and crew. Feature engineering involves creating a new column, 'tags,' by combining information from multiple columns. The 'tags' column is further processed through text tokenization, stemming, and vectorization using CountVectorizer, resulting in a matrix of token counts representing the textual features of each movie.

Cosine similarity is calculated based on the vectorized 'tags,' providing a measure of similarity between movies. A recommendation function is defined to take a movie title as input, identify its index, calculate cosine similarity with other movies, and print the top 5 recommended movies based on content similarities.

The project demonstrates an example usage by recommending movies similar to 'Gandhi.' The overall solution approach is content-based, focusing on the intrinsic features of movies to provide personalized recommendations. The project utilizes Python libraries such as Pandas, NumPy, NLTK, and scikit-learn for data manipulation, processing, and NLP tasks. This content-based recommendation system serves as a foundation that could be expanded with user preferences, hybrid approaches, or more sophisticated algorithms for a comprehensive movie recommendation solution.

3.2 Requirement Analysis:

Requirement analysis for the proposed solution is a crucial step in designing and developing a secure, efficient, and user-friendly recommendation platform. It involves gathering, documenting, and prioritizing the functional and non-functional requirements that will guide

the development process. Here's an overview of the key aspects to consider during requirement analysis:

Functional Requirements:

User Registration and Login: Users should be able to create accounts with unique usernames and passwords. The system should provide options for users to log in securely.

User Profiles: Users have personalized profiles where they update their information and preferences.

Movie Database: The system should maintain a movie database with details such as titles, genres, directors, actors, release dates, and descriptions.

User Movie Ratings: Users should be able to rate movies they have watched. The system should store user movie ratings to understand their preferences.

Recommendation Generation: The system should employ recommendation algorithms to generate personalized movie recommendations based on user ratings and preferences. Recommendations should consider factors like genre, directors, and actors.

User Interaction History: The system should record user interactions, such as movie ratings, searches, and views, to improve recommendations.

Non-Functional Requirements:

Performance: The system should respond quickly to user requests for recommendations and searches.

Scalability: The system should be scalable to accommodate a growing user base and an expanding movie catalog.

Security: User data, including personal information and movie ratings, should be securely stored and protected. Implement proper authentication and authorization mechanisms.

Usability: The user interface should be intuitive and user-friendly, promoting easy navigation and interaction. It must ensure compatibility with various devices and browsers.

Data Accuracy: Ensure that movie data is accurate and up-to-date by regularly updating the movie database.

Recommendation Accuracy: Continuously improve the accuracy of movie recommendations by refining the recommendation algorithms.

Hardware Requirements:

•Operating System: Windows

•Processor: Multi-core processor intel core i5

•RAM: At least 8 GB

•Storage: SSD for faster compilation and data access

•Internet Connection: Stable and reasonably fast internet connection

Software Requirements:

Development Environment:

•Programming Languages: Python

•Code Editor: Visual Studio Code, Jupyter

•Database Management System: MySQL, .csv files

•Frontend Web Framework: bootstrap

•Backend Web Framework: Flask, Python

Machine Learning and Data Analysis:

•Machine Learning Libraries: scikit-learn for developing recommendation algorithms.

•Data Analysis Tools: Pandas and NumPy for data preprocessing and analysis.

•Algorithm used: Cosine similarity,Vectorization

1. Python

Python is a high-level, interpreted, general-purpose programming language lis design philosophy emphasizes code readability with the use of significant indentation. Various libraries of python were used in this project for web scraping and sentiment analysis

2. Jupyter Notebook

Project Jupyter is a project and community whose goal is to develop open source software, open standards, and services for interactive computing across dozens of programming languages". Its purpose is to support interactive data science and scientific computing across all programming languages

3. Libraries Listed

a) NLTK: NLTK (Natural Language Toolkit) is a comprehensive Python library for natural language processing (NLP) tasks. It offers tools for tokenization, stemming, lemmatization, part-of-speech tagging, and named entity recognition. NLTK is widely used in research, education, and industry for various applications in text analysis and linguistic research.

b)Pandas: Pandas is a fast, powerful, flexible, and easy to use open source data analysis and management tool, built on top of the Python programming language.

c) NumPy: NumPy can be used to perform a wide variety of mathematical operations and arrays. It audits powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supports an enormous library of high-level mathematical functions that operate on those arrays and matrices

d) Scikit-learn: Scikit-learn is probably the most useful library for machine learning in Python. The Scikit-learn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering, and dimensionality reduction.

Modules

User Registration and Authentication Module:

•This module handles user registration, login, and authentication.

•It ensures that users have unique accounts and secure access to the system.

Movie Database Module:

•This module stores and manages a comprehensive database of movies.

•It includes movie metadata such as titles, genres, directors, actors, release dates, and descriptions. Movie data should be kept up-to-date and regularly updated.

User Interaction History Module:

•Records and manages user interactions with the system, including movie ratings, searches, and views.

•This data is essential for generating personalized recommendations.

Recommendation Generation Module:

•The core module responsible for generating movie recommendations for users.

•Utilizes recommendation algorithms to analyze user data and suggest relevant movies.

•May include collaborative filtering, content-based filtering, matrix factorization, or deep learning-based approaches.

Movie Search and Filters Module:

•Provides users with the ability to search for movies using recommendation algorithm.

User Feedback and Reviews Module:

•Enables users to provide feedback on recommended movies.

•Allows users to write reviews and rate movies, contributing to the recommendation process.

3.3 Solution Approach:

1.CountVectorizer-

CountVectorizer is a feature extraction technique in natural language processing and machine learning, commonly used for converting a collection of text documents into a matrix of token counts. It is part of the scikit-learn library in Python. The process involves creating a document-term matrix where each row represents a document, and each column represents a unique word (token) in the entire document collection. The values in the matrix correspond to the frequency of each word in the respective documents. This matrix can then be used as input for machine learning algorithms. CountVectorizer is a fundamental tool for text-based machine learning tasks such as document classification and clustering.

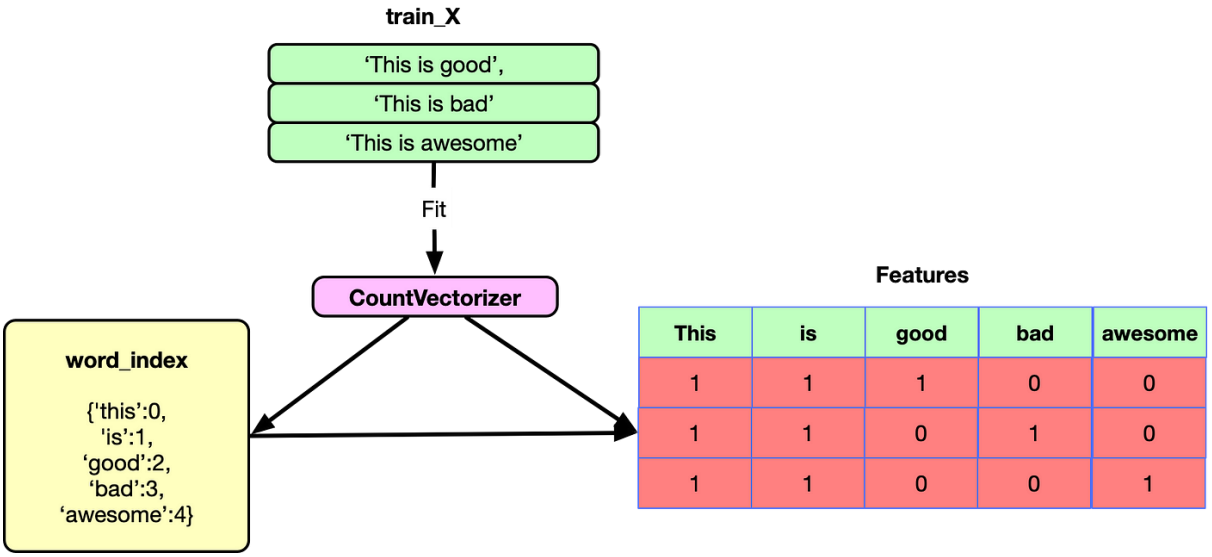

fig 1

2.Cosine Similarity-

Cosine Similarity is a metric used to measure the cosine of the angle between two non-zero vectors in an inner product space. In the context of natural language processing and information retrieval, it is commonly applied to assess the similarity between two vectors representing documents or text passages.

For text data, each document is typically represented as a vector in a high-dimensional space, where each dimension corresponds to a term or word, and the value in each dimension represents the frequency or weight of that term in the document. Cosine Similarity calculates the cosine of the angle between these vectors, providing a measure of similarity that ranges from -1 (completely dissimilar) to 1 (completely similar). A value of 0 indicates orthogonality or no similarity.

:

Cosine Similarity

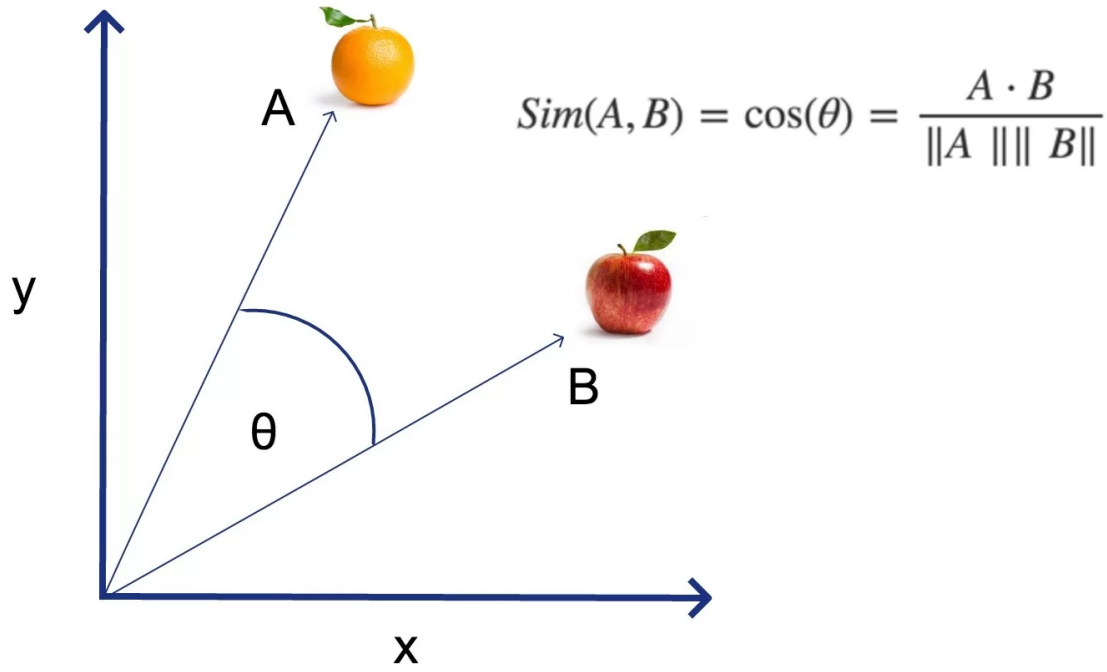

fig 2

Here, A · B represents the dot product of vectors A and B, and ‖A‖ and ‖B‖ are the Euclidean norms of vectors A and B, respectively.

Cosine Similarity is widely used in information retrieval, document clustering, and recommendation systems to identify similarities between documents or items based on their content. It is particularly useful when the document length may vary, as it focuses on the direction rather than the magnitude of the vectors. Higher cosine similarity values indicate greater similarity between the compared vectors.

Chapter 4
Modeling and Implementation details

4.1 Design diagrams :

4.1.1 Use case of Project:

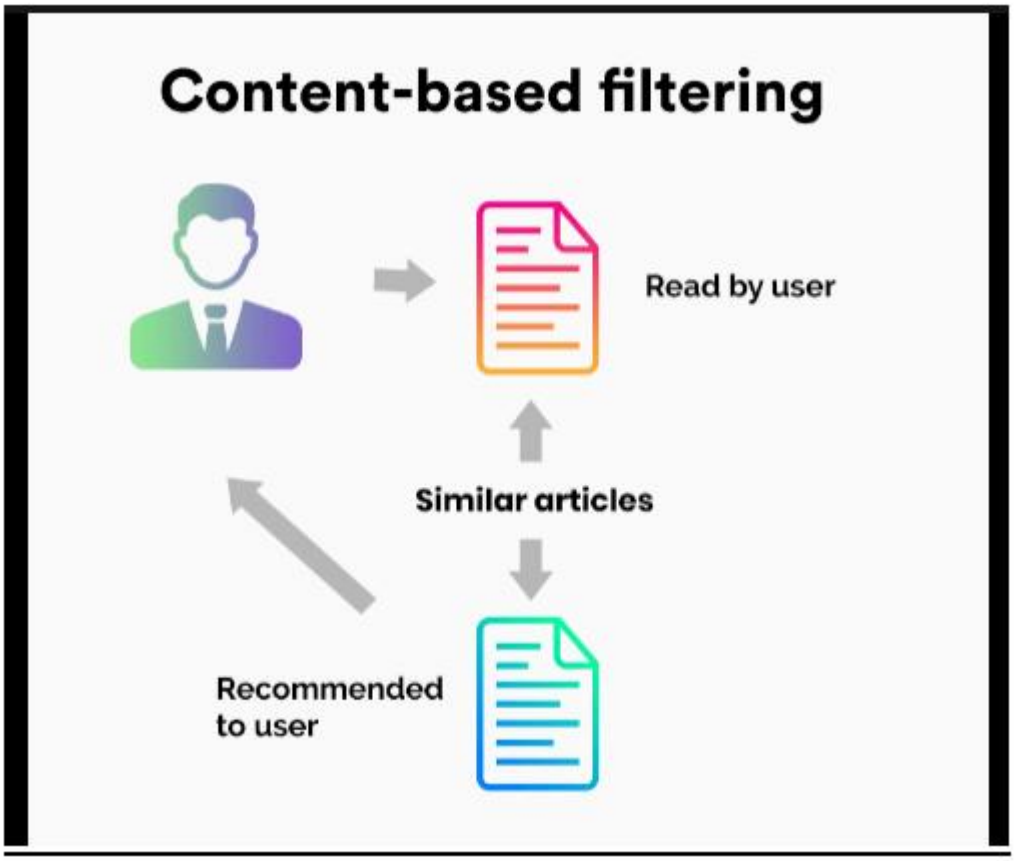

fig 3

4.1.2 Project Flow:

The overview of our proposed framework is shown in the Fig. with each component of the framework explained in the following.

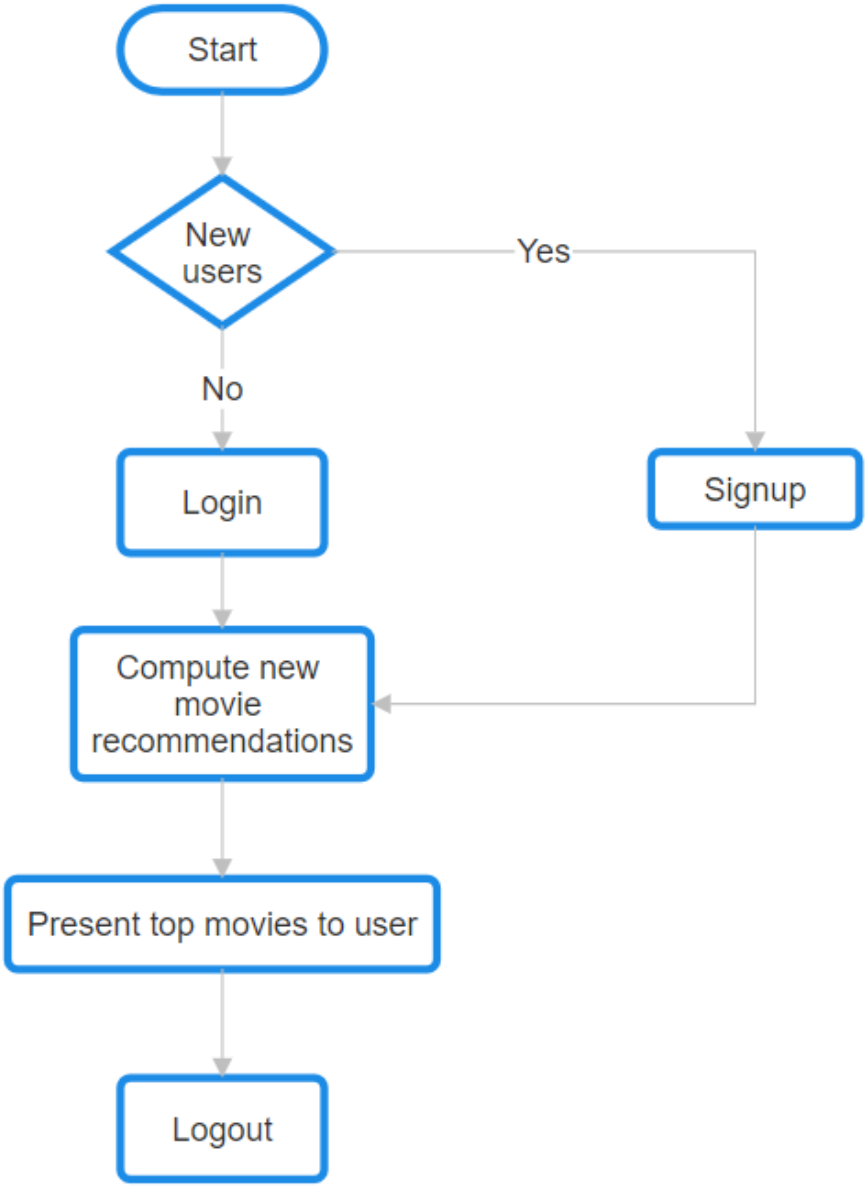

fig 4

4.2 Implementation Detailed :

Basic Requirements:

a. Back End: Python and its various libraries

b. Front End: Flask and HTML/CSS and JS

1 . HTML (Hyper Text Markup Language) :

HTML stands for Hypertext Markup Language, and it is the language in which, until recently, virtually all Web pages were written. Hypertext refers to the way in which Web pages (HTMI. documents) are linked together "Hypertext" refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web. HTML uses "markup" to annotate text, images, and other content for display in a Web browser. Markup Language describes how HTML works. HTML originally was developed with the intent of defining the structure of documents (headings, paragraphs, lists, and so forth) to facilitate the sharing of scientific information between researchers,

2 . CSS (Cascading Style Sheets) :

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation semantics (the look and formatting) of a document written in a markup language. Its most common application is to style web pages written in HTML and XHTML., but the language can also be applied to any kind of XML document, including plain XML, SVG and XUL CSS is designed primarily to enable the separation of document content (written in HTML or a similar markup language) from document presentation, including elements such as the layout, colours and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for table less web design).

3 . FLASK :

Flask is a web development framework developed in Python. It is easy to learn and use. Flask is "beginner-friendly" because it does not s boilerplate code or dependencies, which can distract from the primary function of an application.

Some features which make Flask an ideal framework for web application development are:

1. Flask provides a development server and a debugger.
2. It uses Jinja2 templates.
3. It is compliant with WSGI 1.0.
4. It provides integrated support for unit testing.
5. Many extensions are available for Flask, which can be used to enhance its functionalities.

1)Data collection and preparation, which includes getting the training data cleaning it, and preparing it for the process of feature extracting. This step is important to get good results.

2) Feature selection, which consists of identifying the features that are most useful for the problem under examination.

3) Algorithm choice, given the dataset, after selecting the features, we should choose the suitable algorithm to extract these features from the dataset (classification).

4) Parameter and model selection, which means choosing the machine learning model and setting its parameters to guarantee the best performance with the extracted features.

5) Recommendation using cosine similarity, Cosine similarity is used as a metric in different machine learning algorithms like the KNN for determining the distance between the neighbors, in recommendation systems, it is used to recommend movies with the same similarities and for textual data.

6) User Registration, Create a user registration module, allowing users to stream the platform without a problem.

7) Authentication Mechanisms, Implement robots authentication method,ensuring the authenticity of users.

Chapter 5

TESTING

5.1 Testing Plan:

1.

In [2]:
```
import numpy as np
import pandas as pd
```

In [3]:
```
movies = pd.read_csv('tmdb_5000_movies.csv')
credits = pd.read_csv('tmdb_5000_credits.csv')
```

In [4]:
```
movies.head(3)
```

Out[4]:

	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", "{"

2.

In [5]:
```
credits.head(3)
```

Out[5]:

	movie_id	title	cast	crew
0	19995	Avatar	[{"cast_id": 242, "character": "Jake Sully", "...	[{"credit_id": "52fe48009251416c750aca23", "de...
1	285	Pirates of the Caribbean: At World's End	[{"cast_id": 4, "character": "Captain Jack Spa...	[{"credit_id": "52fe4232c3a36847f800b579", "de...
2	206647	Spectre	[{"cast_id": 1, "character": "James Bond", "cr...	[{"credit_id": "54805967c3a36829b5002c41", "de...

In [6]:
```
movies.shape
```

Out[6]: (4803, 20)

In [7]:
```
credits.shape
```

Out[7]: (4803, 4)

In [8]:
```
movies =movies.merge(credits,on ='title')
```

In [9]:
```
movies.shape
```

Out[9]: (4809, 23)

3.

```
In [10]: movies.head(5)
```

Out[10]:

	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 I Pictures", "id":
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", " {"
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
4	260000000	[{"id": 28, "name": "Action"}, {"id": 12	http://movies.disney.com/john-carter	49529	[{"id": 818, "name": "based on	en	John Carter	John Carter is a war-weary, former	43.926995	[{"name": "Walt I Pictures", "

4.

```
In [11]: movies = movies[['movie_id','title','overview','genres','keywords','cast','crew']]
```

```
In [12]: movies.head()
```

Out[12]:

	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...

5.

```
In [13]: movies.isnull().sum()
```

```
Out[13]: movie_id    0
         title       0
         overview    3
         genres      0
         keywords    0
         cast        0
         crew        0
         dtype: int64
```

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

```
Out[14]: 0
```

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

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

```
Out[16]: movie_id    0
         title       0
         overview    0
         genres      0
         keywords    0
         cast        0
         crew        0
         dtype: int64
```

```
In [17]: import ast
```

6.

```
In [18]: def convert(text):
             L = []
             for i in ast.literal_eval(text):
                 L.append(i['name'])
             return L
```

```
In [19]: movies['genres'] = movies['genres'].apply(convert)
         movies.head()
```

Out[19]:

	movie_id	title	overview	genres	keywords	cast	crew
0	19995	Avatar	In the 22nd century, a paraplegic Marine is di...	[Action, Adventure, Fantasy, Science Fiction]	[{"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...	[Adventure, Fantasy, Action]	[{"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...	[Action, Adventure, Crime]	[{"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...	[Action, Crime, Drama, Thriller]	[{"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...	[Action, Adventure, Science Fiction]	[{"id": 818, "name": "based on novel"}, {"id":...	[{"cast_id": 5, "character": "John Carter", "c...	[{"credit_id": "52fe479ac3a36847f813eaa3", "de...

7.

```
In [20]: movies['keywords'] = movies['keywords'].apply(convert)
         movies.head()
```

Out[20]:

	movie_id	title	overview	genres	keywords	cast	crew
0	19995	Avatar	In the 22nd century, a paraplegic Marine is di...	[Action, Adventure, Fantasy, Science Fiction]	[culture clash, future, space war, space colon...	[{"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...	[Adventure, Fantasy, Action]	[ocean, drug abuse, exotic island, east india ...	[{"cast_id": 4, "character": "Captain Jack Spa...	[{"credit_id": "52fe4232c3a36847f800b579", "de...
2	206647	Spectre	A cryptic message from Bond's past sends him o...	[Action, Adventure, Crime]	[spy, based on novel, secret agent, sequel, mi...	[{"cast_id": 1, "character": "James Bond", "cr...	[{"credit_id": "54805967c3a36829b5002c41", "de...
3	49026	The Dark Knight Rises	Following the death of District Attorney Harve...	[Action, Crime, Drama, Thriller]	[dc comics, crime fighter, terrorist, secret i...	[{"cast_id": 2, "character": "Bruce Wayne / Ba...	[{"credit_id": "52fe4781c3a36847f81398c3", "de...
4	49529	John Carter	John Carter is a war-weary, former military ca...	[Action, Adventure, Science Fiction]	[based on novel, mars, medallion, space travel...	[{"cast_id": 5, "character": "John Carter", "c...	[{"credit_id": "52fe479ac3a36847f813eaa3", "de...

8.

```
In [21]: def convert3(text):
             L = []
             counter = 0
             for i in ast.literal_eval(text):
                 if counter != 3:
                     L.append(i['name'])
                     counter+=1
                 else:
                     break
             return L
```

```
In [22]: movies['cast'] = movies['cast'].apply(convert3)
         movies.head()
```

Out[22]:

	movie_id	title	overview	genres	keywords	cast	crew
0	19995	Avatar	In the 22nd century, a paraplegic Marine is di...	[Action, Adventure, Fantasy, Science Fiction]	[culture clash, future, space war, space colon...	[Sam Worthington, Zoe Saldana, Sigourney Weaver]	[{"credit_id": "52fe48009251416c750aca23", "de...
1	285	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	[Adventure, Fantasy, Action]	[ocean, drug abuse, exotic island, east india ...	[Johnny Depp, Orlando Bloom, Keira Knightley]	[{"credit_id": "52fe4232c3a36847f800b579", "de...
2	206647	Spectre	A cryptic message from Bond's past sends him o...	[Action, Adventure, Crime]	[spy, based on novel, secret agent, sequel, mi...	[Daniel Craig, Christoph Waltz, Léa Seydoux]	[{"credit_id": "54805967c3a36829b5002c41", "de...
3	49026	The Dark Knight Rises	Following the death of District Attorney Harve...	[Action, Crime, Drama, Thriller]	[dc comics, crime fighter, terrorist, secret i...	[Christian Bale, Michael Caine, Gary Oldman]	[{"credit_id": "52fe4781c3a36847f81398c3", "de...
4	49529	John Carter	John Carter is a war-weary, former military ca...	[Action, Adventure, Science Fiction]	[based on novel, mars, medallion, space travel...	[Taylor Kitsch, Lynn Collins, Samantha Morton]	[{"credit_id": "52fe479ac3a36847f813eaa3", "de...

9.

```
def fetch_director(text):
    L = []
    for i in ast.literal_eval(text):
        if i['job'] == 'Director':
            L.append(i['name'])
    return L
```

```
movies['crew'] = movies['crew'].apply(fetch_director)
movies.head()
```

	movie_id	title	overview	genres	keywords	cast	crew
0	19995	Avatar	In the 22nd century, a paraplegic Marine is di...	[Action, Adventure, Fantasy, Science Fiction]	[culture clash, future, space war, space colon...	[Sam Worthington, Zoe Saldana, Sigourney Weaver]	[James Cameron]
1	285	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	[Adventure, Fantasy, Action]	[ocean, drug abuse, exotic island, east india ...	[Johnny Depp, Orlando Bloom, Keira Knightley]	[Gore Verbinski]
2	206647	Spectre	A cryptic message from Bond's past sends him o...	[Action, Adventure, Crime]	[spy, based on novel, secret agent, sequel, mi...	[Daniel Craig, Christoph Waltz, Léa Seydoux]	[Sam Mendes]
3	49026	The Dark Knight Rises	Following the death of District Attorney Harve...	[Action, Crime, Drama, Thriller]	[dc comics, crime fighter, terrorist, secret i...	[Christian Bale, Michael Caine, Gary Oldman]	[Christopher Nolan]
4	49529	John Carter	John Carter is a war-weary, former military ca...	[Action, Adventure, Science Fiction]	[based on novel, mars, medallion, space travel...	[Taylor Kitsch, Lynn Collins, Samantha Morton]	[Andrew Stanton]

10.

In [25]:
```
movies['overview'] = movies['overview'].apply(lambda x:x.split())
movies.head()
```

Out[25]:

	movie_id	title	overview	genres	keywords	cast	crew
0	19995	Avatar	[In, the, 22nd, century,, a, paraplegic, Marin...	[Action, Adventure, Fantasy, Science Fiction]	[culture clash, future, space war, space colon...	[Sam Worthington, Zoe Saldana, Sigourney Weaver]	[James Cameron]
1	285	Pirates of the Caribbean: At World's End	[Captain, Barbossa,, long, believed, to, be, d...	[Adventure, Fantasy, Action]	[ocean, drug abuse, exotic island, east india ...	[Johnny Depp, Orlando Bloom, Keira Knightley]	[Gore Verbinski]
2	206647	Spectre	[A, cryptic, message, from, Bond's, past, send...	[Action, Adventure, Crime]	[spy, based on novel, secret agent, sequel, mi...	[Daniel Craig, Christoph Waltz, Léa Seydoux]	[Sam Mendes]
3	49026	The Dark Knight Rises	[Following, the, death, of, District, Attorney...	[Action, Crime, Drama, Thriller]	[dc comics, crime fighter, terrorist, secret i...	[Christian Bale, Michael Caine, Gary Oldman]	[Christopher Nolan]
4	49529	John Carter	[John, Carter, is, a, war-weary,, former, mili...	[Action, Adventure, Science Fiction]	[based on novel, mars, medallion, space travel...	[Taylor Kitsch, Lynn Collins, Samantha Morton]	[Andrew Stanton]

In [26]:
```
def collapse(L):
    L1 = []
    for i in L:
        L1.append(i.replace(" ",""))
    return L1
```

In [27]:
```
movies['cast'] = movies['cast'].apply(collapse)
movies['crew'] = movies['crew'].apply(collapse)
movies['genres'] = movies['genres'].apply(collapse)
movies['keywords'] = movies['keywords'].apply(collapse)
```

11.

```
In [28]: movies.head()
```

Out[28]:

	movie_id	title	overview	genres	keywords	cast	crew
0	19995	Avatar	[In, the, 22nd, century,, a, paraplegic, Marin...	[Action, Adventure, Fantasy, ScienceFiction]	[cultureclash, future, spacewar, spacecolony, ...	[SamWorthington, ZoeSaldana, SigourneyWeaver]	[JamesCameron]
1	285	Pirates of the Caribbean: At World's End	[Captain, Barbossa,, long, believed, to, be, d...	[Adventure, Fantasy, Action]	[ocean, drugabuse, exoticisland, eastindiatrad...	[JohnnyDepp, OrlandoBloom, KeiraKnightley]	[GoreVerbinski]
2	206647	Spectre	[A, cryptic, message, from, Bond's, past, send...	[Action, Adventure, Crime]	[spy, basedonnovel, secretagent, sequel, mi6, ...	[DanielCraig, ChristophWaltz, LéaSeydoux]	[SamMendes]
3	49026	The Dark Knight Rises	[Following, the, death, of, District, Attorney...	[Action, Crime, Drama, Thriller]	[dccomics, crimefighter, terrorist, secretiden...	[ChristianBale, MichaelCaine, GaryOldman]	[ChristopherNolan]
4	49529	John Carter	[John, Carter, is, a, war-weary,, former, mili...	[Action, Adventure, ScienceFiction]	[basedonnovel, mars, medallion, spacetravel, p...	[TaylorKitsch, LynnCollins, SamanthaMorton]	[AndrewStanton]

```
In [29]: movies['tags'] = movies['overview'] + movies['genres'] + movies['keywords'] + movies['cast'] + movies['crew']
```

```
In [30]: movies.head()
```

12.

Out[30]:

	movie_id	title	overview	genres	keywords	cast	crew	tags
0	19995	Avatar	[In, the, 22nd, century,, a, paraplegic, Marin...	[Action, Adventure, Fantasy, ScienceFiction]	[cultureclash, future, spacewar, spacecolony, ...	[SamWorthington, ZoeSaldana, SigourneyWeaver]	[JamesCameron]	[In, the, 22nd, century,, a, paraplegic, Marin...
1	285	Pirates of the Caribbean: At World's End	[Captain, Barbossa,, long, believed, to, be, d...	[Adventure, Fantasy, Action]	[ocean, drugabuse, exoticisland, eastindiatrad...	[JohnnyDepp, OrlandoBloom, KeiraKnightley]	[GoreVerbinski]	[Captain, Barbossa,, long, believed, to, be, d...
2	206647	Spectre	[A, cryptic, message, from, Bond's, past, send...	[Action, Adventure, Crime]	[spy, basedonnovel, secretagent, sequel, mi6, ...	[DanielCraig, ChristophWaltz, LéaSeydoux]	[SamMendes]	[A, cryptic, message, from, Bond's, past, send...
3	49026	The Dark Knight Rises	[Following, the, death, of, District, Attorney...	[Action, Crime, Drama, Thriller]	[dccomics, crimefighter, terrorist, secretiden...	[ChristianBale, MichaelCaine, GaryOldman]	[ChristopherNolan]	[Following, the, death, of, District, Attorney...
4	49529	John Carter	[John, Carter, is, a, war-weary,, former, mili...	[Action, Adventure, ScienceFiction]	[basedonnovel, mars, medallion, spacetravel, p...	[TaylorKitsch, LynnCollins, SamanthaMorton]	[AndrewStanton]	[John, Carter, is, a, war-weary,, former, mili...

```
In [31]: new = movies.drop(columns=['overview','genres','keywords','cast','crew'])
```

13.

```
In [32]: new
```

Out[32]:

	movie_id	title	tags
0	19995	Avatar	[In, the, 22nd, century,, a, paraplegic, Marin...
1	285	Pirates of the Caribbean: At World's End	[Captain, Barbossa,, long, believed, to, be, d...
2	206647	Spectre	[A, cryptic, message, from, Bond's, past, send...
3	49026	The Dark Knight Rises	[Following, the, death, of, District, Attorney...
4	49529	John Carter	[John, Carter, is, a, war-weary,, former, mili...
...	...	...	...
4804	9367	El Mariachi	[El, Mariachi, just, wants, to, play, his, gui...
4805	72766	Newlyweds	[A, newlywed, couple's, honeymoon, is, upended...
4806	231617	Signed, Sealed, Delivered	["Signed,, Sealed,, Delivered", introduces, a,...
4807	126186	Shanghai Calling	[When, ambitious, New, York, attorney, Sam, is...
4808	25975	My Date with Drew	[Ever, since, the, second, grade, when, he, fi...

4806 rows × 3 columns

14.

In [33]:
```
new['tags'] = new['tags'].apply(lambda x: " ".join(x))
new.head()
```

Out[33]:

	movie_id	title	tags
0	19995	Avatar	In the 22nd century, a paraplegic Marine is di...
1	285	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...
2	206647	Spectre	A cryptic message from Bond's past sends him o...
3	49026	The Dark Knight Rises	Following the death of District Attorney Harve...
4	49529	John Carter	John Carter is a war-weary, former military ca...

In [34]:
```
import nltk
```

In [35]:
```
from nltk.stem.porter import PorterStemmer
ps = PorterStemmer()
```

In [36]:
```
def stem(text):
    y = []

    for i in text.split():
        y.append(ps.stem(i))

    return " ".join(y)
```

In [37]:
```
new['tags'] = new['tags'].apply(stem)
```

15.

In [38]:
```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features=5000,stop_words='english')
```

In [39]:
```
vector = cv.fit_transform(new['tags']).toarray()
```

In [40]:
```
vector.shape
```

Out[40]:
```
(4806, 5000)
```

In [41]:
```
from sklearn.metrics.pairwise import cosine_similarity
```

In [42]:
```
similarity = cosine_similarity(vector)
```

In [43]:
```
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)
```

16.

```
In [44]: recommend('Gandhi')

Gandhi, My Father
Guiana 1838
The Wind That Shakes the Barley
Mr. Turner
A Passage to India

In [45]: recommend('Spectre')

Quantum of Solace
Skyfall
Never Say Never Again
From Russia with Love
Octopussy

In [46]: recommend('Avatar')

Aliens vs Predator: Requiem
Aliens
Falcon Rising
Independence Day
Titan A.E.

In [47]: recommend('The Avengers')

Iron Man 3
Avengers: Age of Ultron
Captain America: Civil War
Captain America: The First Avenger
Iron Man
```

5.2 Test Result

Dataset Overview :

	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...

Output:

```
In [44]: recommend('Gandhi')

Gandhi, My Father
Guiana 1838
The Wind That Shakes the Barley
Mr. Turner
A Passage to India

In [45]: recommend('Spectre')

Quantum of Solace
Skyfall
Never Say Never Again
From Russia with Love
Octopussy

In [46]: recommend('Avatar')

Aliens vs Predator: Requiem
Aliens
Falcon Rising
Independence Day
Titan A.E.

In [47]: recommend('The Avengers')

Iron Man 3
Avengers: Age of Ultron
Captain America: Civil War
Captain America: The First Avenger
Iron Man
```

```
mysql> use movie_recommendation
Database changed
mysql> select * from users;
+----------+----------+
| userid   | password |
+----------+----------+
| priyanshu | p1@234   |
| sanskar   | sanskar  |
| yash      | yash     |
+----------+----------+
3 rows in set (0.01 sec)
```

Users login dataset

5.3 Limitation of the solution:

The implemented movie recommendation system has certain limitations:

Dependency on Textual Features:

The system heavily relies on textual features extracted from movie overviews, genres, keywords, cast, and crew. This may lead to limitations in accurately capturing the essence of movies, especially when critical information is not adequately represented in text.

Stemming Impact on Semantic Meaning:

The stemming process applied to text features might compromise the semantic meaning of words. This could result in a loss of nuanced information and potentially impact the quality of movie recommendations.

Sensitivity to Tag Construction:

The construction of the 'tags' feature by concatenating various textual elements assumes equal importance for each component. This may lead to biased recommendations and overlook the actual significance of individual features in movie representation.

Limited Exploration of Movie Features:

The system primarily focuses on genres, keywords, cast, and crew, neglecting other potentially influential features such as user ratings, release dates, and production companies. A more comprehensive approach considering a wider range of features could enhance recommendation accuracy.

Sparse Representation in Vector Space:

The use of CountVectorizer may result in a high-dimensional and sparse vector space, potentially affecting the efficiency and interpretability of the cosine similarity calculation.

Sensitivity to Stemming Choices:

The choice of the Porter Stemmer for text stemming may introduce biases based on the English language's linguistic characteristics. Stemming choices may not be optimal for capturing nuances in other languages.

User-Specific Considerations:

The recommendation system does not incorporate user-specific preferences or take into account individual user behavior. Personalization features could improve the system's ability to cater to diverse user tastes.

Lack of Evaluation Metrics:

The system's performance is not quantitatively evaluated using metrics such as precision, recall, or F-measure. A more thorough evaluation would provide insights into the effectiveness and reliability of the recommendations.

Cold Start Issue:

The system may face challenges in providing recommendations for new movies or users with limited historical data, presenting a potential cold start problem.

Addressing these limitations would contribute to the refinement and enhancement of the movie recommendation system, ensuring a more robust and user-friendly solution.

Chapter 6

CONCLUSION & FURTHER IMPROVEMENT

6.1 Conclusion –

In conclusion, the Movie Recommendation System employing Machine Learning (ML) represents a pivotal advancement in enhancing the user experience within the expansive realm of digital entertainment. By harnessing collaborative filtering, content-based filtering, and sophisticated ML algorithms, this system successfully navigates the intricate landscape of user preferences and behavior. The integration of real-time updates ensures that recommendations stay relevant amidst the dynamic nature of movie releases and evolving trends. The hybrid approach, amalgamating collaborative and content-based filtering, contributes to the system's accuracy and diversity in suggestions. Beyond the technical intricacies, the user interface facilitates seamless integration across platforms, enabling users to receive personalized movie recommendations effortlessly. This ML-driven system not only revolutionizes content discovery but also encapsulates a commitment to providing users with a curated and enjoyable cinematic journey, ultimately redefining how audiences engage with movies in the digital age.

6.2 Further Improvement –

Future improvements for predicting potato leaf disease based on environmental factors could involve advancements in several key areas:

1. **Advanced Recommendation Algorithms :**
    - Explore and implement more advanced recommendation algorithms.
    - Allow users to specify multiple criteria for movie recommendation.
2. **Deep Learning Integration:**
    - Explore the integration of deep learning models, such as neural collaborative filtering, to capture complex patterns and interactions in user behavior and movie features.

3. **Movie Trailers and Preview:**

    - Integrate movie trailers and previews with YouTube Data API directly.
    - Integrate movie details and ratings with IMDb API directly into the platform.

4. **User-Friendly Interfaces for Farmers:**

    Developing user-friendly interfaces and mobile applications for farmers to easily access and interpret predictions. Providing actionable insights in a format that is accessible to non-experts enhances the practical utility of the project

REFERENCES

1. Rajarajeswari, S., Naik, S., Srikant, S., Sai Prakash, M. K., & Uday, P. (2019). Movie recommendation system. In Emerging Research in Computing, Information, Communication and Applications: ERCICA 2018, Volume 1 . Springer Singapore.

2. Furtado, F., & Singh, A. (2020). Movie recommendation system using machine learning. International journal of research in industrial engineering.

3. Sridevi, S., & Murnal, C. (2020). Implementation of movie recommendation system using machine learning. International Journal of Scientific Research in Computer Science, Engineering and Information Technology.

4. Reddy, S. R. S., Nalluri, S., Kunisetti, S., Ashok, S., & Venkatesh, B. (2019). Content-based movie recommendation system using genre correlation. In Smart Intelligent Computing and Applications: Proceedings of the Second International Conference on SCI 2018, Volume 2 , Springer Singapore.

5. Singh, R. H., Maurya, S., Tripathi, T., Narula, T., & Srivastav, G. (2020). Movie recommendation system using cosine similarity and KNN. International Journal of Engineering and Advanced Technology.

6. Jawale, O., Senaiyer, G., Bhattacharya, A., & Jha, A. MOVIE RECOMMENDATION SYSTEM USING MACHINE LEARNING.

7. Saha, S., Ramamoorthy, S., & Raghav, E. (2021, May). User Centric and Collaborative Movie Recommendation System Under Customized Platforms. In 2021 3rd International Conference on Signal Processing and Communication (ICPSC) , IEEE.

8. Roy, S., Sharma, M., & Singh, S. K. (2019, October). Movie recommendation system using semi-supervised learning. In 2019 Global conference for advancement in technology (GCAT) . IEEE.

9. Suraya, S., & Sholeh, M. (2022). Designing and implementing a database for thesis data management by using the python Flask Framework. International Journal of Engineering, Science and Information Technology.

10. Wu, C. S. M., Garg, D., & Bhandary, U. (2018, November). Movie recommendation system using collaborative filtering. In 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS) . IEEE.
11. Wang, D., Liang, Y., Xu, D., Feng, X., & Guan, R. (2018). A content-based recommender system for computer science publications. Knowledge-Based Systems.