

HYBRID MOVIE RECOMMENDATION SYSTEM

Submitted in partial fulfillment of the requirements for
the award of

Bachelor of Engineering degree in Computer Science and Engineering

by

NAGA SAMHITHA ILLA (38110346)

KOVI PRATHYUSHA(38110269)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING SCHOOL OF COMPUTING

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade “A” by NAAC

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600 119
MARCH – 2022**



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **NAGA SAMHITHA ILLA (38110346)**, **KOVI PRATHYUSHA (38110269)** who carried out the project entitled "**HYBRID MOVIE RECOMMENDATION SYSTEM**" under my supervision from November 2021 to March 2022.

InternalGuide

(Dr. M .Selvi ME.,Ph.D)

Head of the Department

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

I **Naga Samhitha Illa(Reg No:38110346)** and **Kovi Prathyusha (Reg No: 38110269)** hereby declare that the Project Report entitled “**HYBRID MOVIE RECOMMENDATION SYSTEM** ” done by us under the guidance of **Dr. M. Selvi ME.,Ph.D** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in 2018-2022.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D, Dean**, School of Computing **Dr. L. Lakshmanan M.E., Ph.D. ,** and **Dr.S.Vigneshwari M.E., Ph.D. Heads** of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.M.Selvi ME.,Ph.D** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project

ABSTRACT

Every year new movies are released with a varied story-line or a genre which could be of potential interest to viewers. Various online movie or video streaming platforms can keep the customers engaged by recommending movies of the viewer's preference. Hybrid Recommendation System for Movies uses the combination of collaborative and content based filtering in the context of web-based recommender systems. Recommendation systems constitute a specific type of information filtering that attempt to present items according to the interests expressed by a user , in order to reduce the effect of such dependencies. Our project proposes a hybrid recommendation system with sentiment analysis of movie tweets. The movie tweets have been collected from micro blogging websites to understand the current trends and user response of the movie. Through different investigations, we will show the ability to expand utilization, just as the qualities of how to foresee blended longings. Ideas for particular kinds of channel data endeavor to decide factors dependent on the business' advantages to limit the effect of protection. Our venture requires an arrangement of correspondence and passionate examination of tweets. The film's tweets have been gathered on a microblog webpage to stay aware of the most recent patterns and arrangements from film clients. Leading examination in the state files gives positive outcomes.

Table of Content

Chapter No.	TITLE	Page No.
	ABSTRACT	v
1	INTRODUCTION	1
	1.1 RELEVANCE OF THE PROJECT	1
	1.2 PROBLEM STATEMENT	2
	1.3 OBJECTIVES OF THE PROJECT	2
	1.4 SCOPE OF THE PROJECT	2
2	LITERATURE REVIEW	3
	2.1 RELATED WORK	3
3	METHODOLOGY	5
	3.1 EXISTING WORK	5
	3.2 PROPOSED WORK	5
	3.3 SYSTEM REQUIREMENT SPECIFICATION	6
	3.3.1 HARDWARE REQUIREMENTS	6
	3.3.2 SOFTWARE REQUIREMENTS	6
	3.4 SOFTWARE SPECIFICATION	7
	3.4.1 FRONT-END SPECIFICATION	7
	3.4.2 BACK-END SPECIFICATION	12
	3.5 SYSTEM ANALYSIS AND DESIGN	15
	3.5.1 SYSTEM ARCHITECTURE	15
	3.5.2 ACTIVITY DIAGRAM	16
	3.5.3 DATA FLOW	16

3.6	RECOMMENDER SYSTEM TYPES AND TECHNIQUES	17
3.6.1	ENTITIES IN RECOMMENDATION SYSTEM	17
3.6.2	COLLABORATIVE FILTERING	20
3.6.3	CONTENT-BASED RECOMMENDATION	21
3.6.4	HYBRID RECOMMENDATION AND APPROACH	23
3.7	DATA ANALYSIS	25
3.7.1	DATA COLLECTION	25
3.7.2	DATA PRE-PROCESSING	27
3.8	IMPLEMENTATION	27
3.8.1	COSINE SIMILARITY	27
3.8.2	KNN	28
3.8.3	SVD	29
4	RESULTS AND DISCUSSION	30
5	CONCLUSION AND FUTURE SCOPE	32
5.1	CONCLUSION	32
5.2	FUTURE SCOPE	32
	APPENDICES	34
	A.SOURCE CODE	34
	B.SCREENSHOTS	39
	C.PUBLICATION WITH PLAGARISM REPORT	42

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
3.1	ARCHITECTURE OF HYBRID APPROACH	15
3.2	ACTIVITY DIAGRAM	16
3.3	DATA FLOW DIAGRAM	16
3.4	EXAMPLE OF TAXONOMY OF THE RECOMMENDATION SYSTEMS	19
3.5	RECOMMENDATION TECHNIQUE	20
3.6	OVERVIEW OF RECOMMENDATION SYSYEMS	21
3.7	CLASSIFICATION ABOUT NEW SAMPLE	28
B	SCREENSHOTS	39

LIST OF TABLES

TABLE NO	TITLE	PAGE NO
3.1	DIFFERENT HTTP METHODS	14
4.1	COMPARISON BETWEEN THE THREE APPLICATIONS	31

CHAPTER 1

INTRODUCTION

1.1 RELEVANCE OF THE PROJECT

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

Movies are a part and parcel of life. There are different types of movies like some for entertainment, some for educational purposes, some are animated movies for children, and some are horror movies or action films. Movies can be easily differentiated through their genres like comedy, thriller, animation, action etc. Other way to distinguish among movies can be either by releasing year, language, director etc. Watching movies online, there are a number of movies to search in our most liked movies .

Movie Recommendation Systems helps us to search our preferred movies among all of these different types of movies and hence reduce the trouble of spending a lot of time searching our favourable movies. So, it requires that the movie recommendation system should be very reliable and should provide us with the recommendation of movies which are exactly same or most matched with our preferences. A large number of companies are making use of recommendation systems to increase user interaction and enrich a user's shopping experience. Recommendation systems have several benefits, the most important being customer satisfaction and revenue. Movie Recommendation system is very powerful and important system. But, due to the problems associated with pure collaborative approach, movie recommendation systems also suffers with poor recommendation quality and scalability issues.

1.2 PROBLEM STATEMENT

The goal of the project is to recommend a movie to the user. Providing related content out of relevant and irrelevant collection of items to users of online service providers.

1.3 OBJECTIVE OF THE PROJECTS

- Improving the Accuracy of the recommendation system.
- Improve the Quality of the movie Recommendation system.
- Improving the Scalability.
- Enhancing the user experience

1.4 SCOPE OF THE PROJECT

The objective of this project is to provide accurate movie recommendations to users. The goal of the project is to improve the quality of movie recommendation system, such as accuracy, quality and scalability of system than the pure approaches. This is done using Hybrid approach by combining content based filtering and collaborative filtering, To eradicate the overload of the data, recommendation system is used as information filtering tool in social networking sites. Hence, there is a huge scope of exploration in this field for improving scalability, accuracy and quality of movie recommendation systems Movie Recommendation system is very powerful and important system. But, due to the problems associated with pure collaborative approach, movie recommendation systems also suffers with poor recommendation quality and scalability issues.

CHAPTER 2

LITERATURE SURVEY

2.1.RELATED WORK

- The work of Nandagawali and Patil [1] provides a solution for the various recommender systems by using collaborative filtering algorithms with the community based user domain model. The main purpose is to satisfy the customer's product needs by providing them recommendation based on products. It deals with the Amazon dataset. Better hardware configuration may result in better output.
- The work of Jatinder et al., [2] deals with the review of several techniques that are used by a system for recommending electronic products. They have also worked with the Amazon dataset. Time complexity of the whole model is pretty high, making it less convenient.
- In the work of Ekaspur and Pashupatimath, [3] a formal representation of social network is present where text mining is taken as a perspective. A framework is proposed that can recommend friend using an efficient Algorithm. Their work is based on Facebook data. The notable points in their work can be that the accuracy is high, and works fine with any size of data. Analysis of the classes and its classification is also highly précised. But it lacks to meet the standard when it comes to time of execution.
- Haruna, et al., [4] in their work, have proposed a research paper recommender system that transforms all the recommending papers into a paper-citation relations matrix. The key points that makes their model better can be faster results. Their algorithm which is Brute Force based, but fails to meet the compatibility in real time application as it will be inefficient for very large dataset, as the number of rows to column ratio will increase abnormally.
- In the research work carried out by Kumar et al., [5] they have proposed a movie recommendation system named MOVREC. The system sorts the ratings by implementing K-means clustering algorithm. The pros of the system include

that the time taken is less, and almost every time the result of the classification of data is accurate [21][25]. But the cons can be that creating classes amongst closely related data which may belong to the same cluster can be redundant and misleading.

- According to the work of Chen et al. [6], they have incorporated Singular Value Decomposition (SVD) in user based collaborative filtering technique for movie recommendation system. Although, they have got good accuracy with their model, but additional user or item features can be applied along with a larger dataset for better preparation of model.
- The work of Reddy, et al. [7] deals with genre correlation-oriented content-based movie recommendation system, that only focusses on the genre of a movie. On the contrary, for any person, choice might not be completely based on one genre.
- In [8], C. S. M. Wu, D. Garg, and U. Bhandary proposed a recommendation system using collaborative filtering where a user's rating is used to suggest the list. The authors have used the Apache Mahout framework and essentially compared the performances and efficiency of user-based & item-based recommendations.
- In [9], R. E. Nakhli, H. Moradi, and M. A. Sadeghi proposed the percentage view approach for recommending movies to the users, it finds relevant movies for the customer and then compares the performance with a random movie recommendation system for showing the accuracy of the project.
- In [10], a content-based recommendation system is proposed by H. W. Chen, Y. L. Wu, M. K. Hor, and C. Y. Tang using neural networks. In recent years, these are top topics for the researchers to work on when they want to build a movie recommendation system.

CHAPTER 3

METHODOLOGY

3.1 EXISTING WORK

In the existing system, collaborative filtering and content based filtering have been used. Collaborative filtering system maintains a database of many users ratings of a variety of items. It makes use of user data, ignoring content/item data. Content based filtering uses only the item data maintaining a profile for each item. Each user assumed to operate independently . No need for data on other users.

Disadvantages of existing work:

Disadvantages of Collaborative filtering:

- Cold start.
- Sparsity.
- First rater.
- Popularity bias.

Disadvantages of content based filtering:

- Cannot filter items on some assessment of quality, style or viewpoint because of lack of consideration of other people.
- Absence of personal recommendations.
- No serendipitous.

3.2 PROPOSED WORK

In the proposed hybrid recommender system takes advantage from both the representation of the content as well as the similarities among users. The intuition behind this technique is to use a content-based predictor to fill the user-rating matrix that is sparsely distributed.

A web crawler is used to download necessary movie content for our dataset. After the preprocessing the movie content database is stored. The dataset consists of a user-rating matrix.

Content-based predictions are used to train each user-rating vector in the user-rating matrix and convert it into a pseudo rating matrix which combines actual rating with the predicted ratings. Collaborative filtering is then applied to this full pseudo user-rating matrix to make recommendation for an active user. The movie tweets have been collected from microblogging websites to understand the current trends and user response of the movie.

3.3 SYSTEM REQUIREMENT SPECIFICATION

This involves both the hardware and software requirements needed for the project and detailed explanation of the specifications.

3.3.1 Hardware Requirements

- A PC with Windows/Linux OS
- Processor with 1.7-2.4GHz speed
- Minimum of 8gb RAM
- 2gb Graphic card

3.3.2 Software Requirements

- Text Editor (VS-code/WebStorm)
- Anaconda distribution package
- Python libraries

Anaconda distribution: Anaconda is a free and open-source distribution of the Python programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management system and deployment. Package versions are managed by the package management system conda. The anaconda distribution includes data-science packages suitable for Windows, Linux and MacOS.³

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

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

NumPy: NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python.

Pandas: Pandas is one of the most widely used python libraries in data science. It provides high-performance, easy to use structures and data analysis tools. Unlike NumPy library which provides objects for multi-dimensional arrays, Pandas provides in-memory 2d table object called Data frame.

Flask: It is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug.

3.4 SOFTWARE SPECIFICATION

3.4.1 FRONT-END SPECIFICATION

I. HTML:

HTML or Hypertext Markup Language is the standard markup language used to create web pages. HTML is written in the form of HTML elements consisting of *tags* enclosed in angle brackets (like <html>). HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent *empty elements* and so are unpaired, for example . The first tag in a pair is the *start tag*, and the second tag is the *end tag* (they are also called *opening tags* and *closing tags*). Though not always necessary, it is best practice to append a slash to tags which are not paired with a closing tag.

The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language rather than a programming language.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as Java Script which affect the behavior of HTML web pages.

FEATURES of HTML:

- It is easy to learn and easy to use.
- It is platform-independent.
- Images, videos, and audio can be added to a web page.
- Hypertext can be added to text.
- It is a markup language.

Why use HTML?

- It is a simple markup language. Its implementation is easy.
- It is used to create a website.
- Helps in developing fundamentals about web programming.
- Boost professional career.

II. CASCADING STYLE SHEETS (CSS):

CSS is a cornerstone specification of the web and almost all web pages use CSS style sheets to describe their presentation. CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors, 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.

It is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to style web pages and interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. CSS can also allow the same markup page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices. It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed. While the author of a document typically links that document to a CSS file, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified. However if the author or the reader did not link the document to a specific style sheet the default style of the browser will be applied.

Features of CSS:

- Solves a big problem
- Saves a lot of time
- Provide more attributes
- Pages load faster
- Easier Website maintenance
- Multiple device compatibility

Why use CSS?

CSS saves time: You can write CSS once and reuse the same sheet in multiple HTML pages.

Easy Maintenance: To make a global change simply change the style, and all elements in all the webpages will be updated automatically.

Search Engines: CSS is considered a clean coding technique, which means search engines won't have to struggle to "read" its content.

Superior styles to HTML: CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.

Offline Browsing: CSS can store web applications locally with the help of an offline cache. Using this we can view offline websites.

III. JAVASCRIPT:

JavaScript is the scripting language of the Web. All modern HTML pages are using JavaScript. A scripting language is a lightweight programming language. JavaScript code can be inserted into any HTML page, and it can be executed by all types of web browsers. JavaScript is easy to learn.

WHY TO USE JAVASCRIPT:

JavaScript is one of the 3 languages all web developers must learn:

1.61.1 HTML to define the content of web pages

1.61.2 CSS to specify the layout of web pages

1.61.3 JavaScript to specify the behavior of web pages

Example:

```
x = document.getElementById("demo"); //Find the HTML element with id="demo"
x.innerHTML = "Hello JavaScript";      //Change the content of the HTML element
```

document.getElementById() is one of the most commonly used HTML DOM methods.

JAVASCRIPT CODE:

- JavaScript code (or just JavaScript) is a sequence of JavaScript statements.
- Each statement is executed by the browser in the sequence they are written.
- This example will manipulate two HTML elements:
- Example

```
document.getElementById("demo").innerHTML="Hello    Dolly";  
document.getElementById("myDIV").innerHTML="How      are  
you?";
```

JAVASCRIPT PROPERTIES:

- Properties are the values associated with a JavaScript object.
- A JavaScript object is a collection of unordered properties.
- Properties can usually be changed, added, and deleted, but some are read only.

IV. BOOTSTRAP:

WHAT IS BOOTSTRAP?

Bootstrap is the most popular **CSS Framework** for developing responsive and mobile-first websites.

Bootstrap is a potent front-end framework used to create modern websites and web apps. It's open-source and free to use, yet features numerous HTML and CSS templates for UI interface elements such as buttons and forms. Bootstrap also supports JavaScript extensions.

FEATURES of BOOTSTRAP:

Software engineers use Bootstrap for a number of different reasons:

- It is easy to set up and master
- It has a lot of components and a good grid system.
- Styling for many HTML elements ranging from typography to buttons, as well as support of JavaScript plugins, makes it very

flexible.

WHAT ARE THE USES OF BOOTSTRAP:

- Bootstrap is great for creating layouts, as its responsive CSS is designed to conform to different devices.
- It can be employed to ensure consistency, eliminate cross-browser issues, and so on.

3.4.2 BACK-END SPECIFICATION

1. PYTHON:

What is Python?

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

What is Python used for:

- Web development (server-side),
- Software development,
- Mathematics,
- System scripting.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Features of Python:

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.

Python can be treated in a procedural way, an object-oriented way or a functional way.

Example.

```
print("Hello, World!")
```

Output

Hello, World!

2. FLASK Framework

Flask is a web application framework written in Python. Armin Ronacher, who leads an international group of Python enthusiasts named Pocco, develops it. Flask is based on Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects

.

Http protocol is the foundation of data communication in world wide web. Different methods of data retrieval from specified URL are defined in this protocol.

The following table summarizes different http methods :

Table 3.1 Different HTTP methods

Sl.No	Methods & Description
1	GET Sends data in unencrypted form to the server. Most common method.
2	HEAD Same as GET, but without response body
3	POST Used to send HTML form data to server. Data received by POST method is not cached by server.
4	PUT Replaces all current representations of the target resource with the uploaded content.
5	DELETE Removes all current representations of the target resource given by a URL

By default, the Flask route responds to the **GET** requests. However, this preference can be altered by providing methods argument to **route()** decorator.

3.5 SYSTEM ANALYSIS AND DESIGN

3.5.1 system architecture of the proposed system

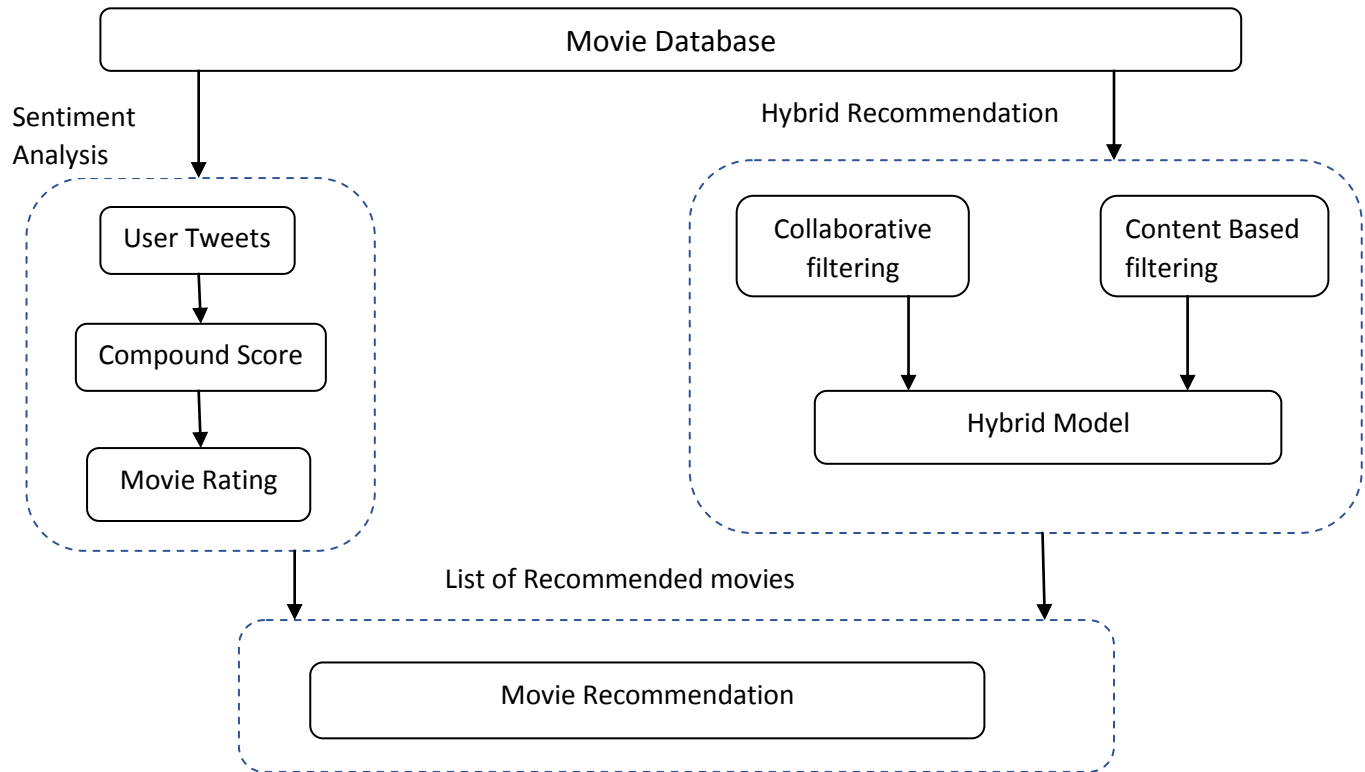


Fig.3.1:Architecture for hybrid approach

For each different individual use different list of movies are recommended ,as user login or enters the user id based on two different approaches used in the project each will recommend the set of movies to the particular user by combining the both the set of movie based on the user the hybrid model will recommend the single list of movie to the user.

3.5.2 Activity diagram :

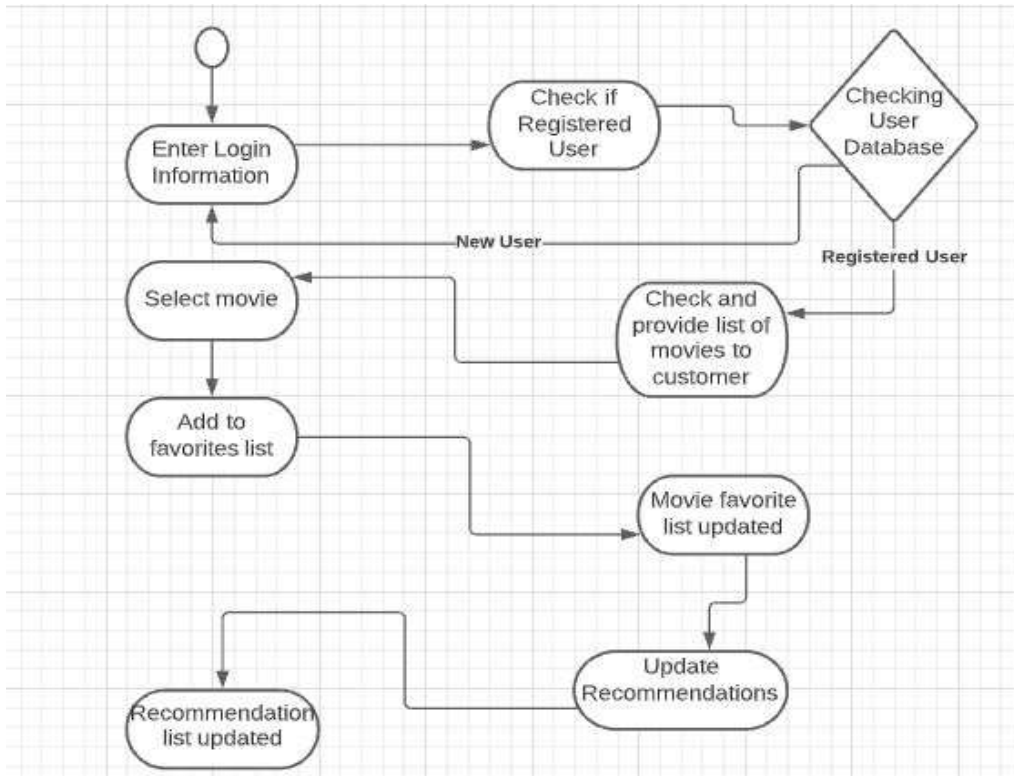


Fig.3.2: activity diagram

Once the user login by entering the userid i.e present in the csv file ranges from 1-5000 the list of movie are recommended to the user

3.5.3 DATA FLOW:

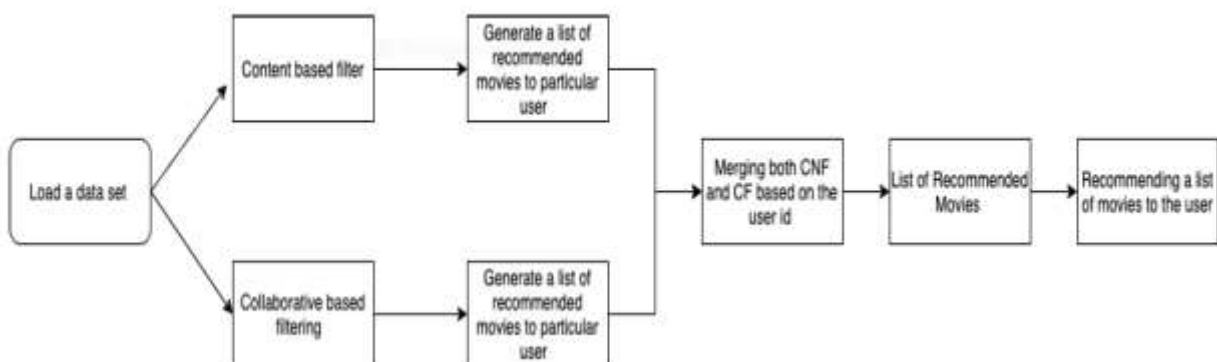


Fig.3.3: data flow diagram

Initially load the data sets that are required to build a model the data set that are required in this project are movies_metadata.csv, ratings_small.csv, users.csv all the data sets are available in the Kaggle.com. Basically, two models are built in this project content based and collaborative filtering each produce a list of movies to a particular user by combining both based on the userid a single final list of movies are recommended to the particular user.

3.6 RECOMMENDER SYSTEM TYPES AND TECHNIQUES

Recommender systems have evolved in response to an apparent need: helping people deal with the huge explosion of information on the internet. Simply put, it was developed to alleviate the problem of information overload. In addition, it has become obvious that it can connect people who share similar interests, and not just with relevant information.

The Recommender System is a system that involves predicting user responses to options. It offers online users suggestions of what their interest might be, based on their past actions such as a history of purchases and/or product searches, clicks, and ratings. The ultimate aim of a RS is to provide a suggestion that is aimed at supporting users in various decision-making processes.

Amazon uses this technique to display to a given user a list of recommended items that may be of interest, drawing information from the user's past preferences and actions. There are recommender engines that work behind the scenes to capture user behavior and recommend selected items based on their earlier actions. Facebook uses the same recommender technique to determine friends to suggest, thereby creating the "people you may know list".

3.6.1 ENTITIES IN RECOMMENDATION SYSTEMS

Common classes of entities to be explained in a RS are underlined and explained below:

- **Item** is a general term used to denote what the system recommends to users.
- RSs collect from **Users** their preference or are inferred by interpreting user action for example clicking a product may serve as an implicit preference. Users have preferences for certain items and these preferences must be separated out of the data. The data itself is represented as a Utility Matrix.
- **Utility Matrix** represents a user-item pair where users rate items on a scale. This rating depicts the degree of preference of that user for that item. It is usually on a 1-5 scale.
- The goal of a recommender system is to predict the blanks in the Utility Matrix.
- Populating the Utility Matrix is a highly important task as it is almost impossible to recommend items without it.
- **Transactions** refer to a recorded interaction between a user and the RS. Transactions are log-like data that store important information generated during the human-computer interaction and which are useful for the recommendation generation algorithm that the system is using. For instance, the transaction log may contain a reference to a selected item by the user and a description of the context (e.g., the user goal/query) for that particular recommendation. If seen, that transaction may also include a direct comment the user has provided, such as the rating for the selected item. Literally, ratings represent the most popular form of transaction data that a RS collects.

These ratings may be collected explicitly or implicitly. In the explicit collection of ratings, the user is asked to rate a document on a pre-defined scale. User actions are recorded and a rating is inferred in implicit ratings.

There are two general approaches to discover the value users place on items:

1. Users can be asked to rate items. The limitations of this approach are based on the fact that :
 - a. Users are generally unwilling to provide responses
 - b. The information may be biased by the fact that it comes from people

2. Inferences can be made from users' behavior. One can infer interest from behavior other than purchasing, for instance, if a user watches a movie on Youtube, previews a book on Amazon, then we can infer that the user "likes" this item.

Specifically, recommender systems have the following components:

- i. background data, the initial information that the system starts with before the recommendation process begins,
- ii. input data, the information required of the user by the system in order to generate a recommendation, and
- iii. an algorithm that combines background and input data to arrive at its suggestions.

According to Robin Burke, he distinguished five techniques of the recommendation (Figure 3.4) according to the type of a background and input data as well as the algorithm that is used to create the suggestions.

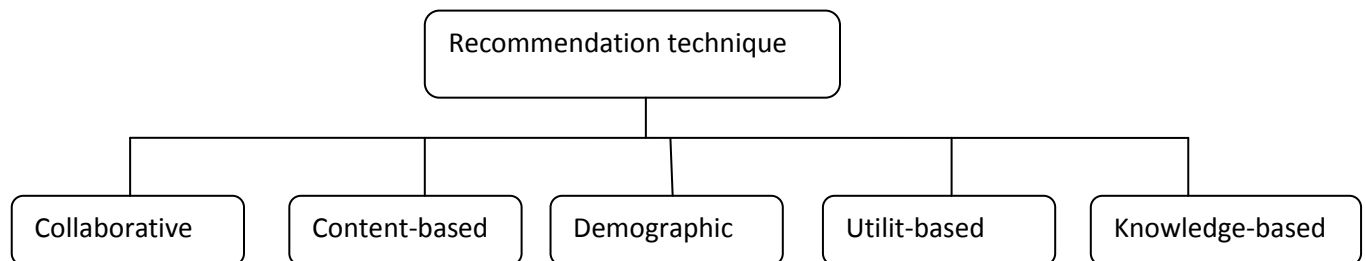


Figure.3.4: The example of taxonomy of the recommender systems

Some other researcher distinguishes three main categories of RSs as follows: Collaborative filtering, Content-based filtering, and Hybrid methods (Figure 2.2). For more general information and examples of these techniques, see F. Ricci et al. This thesis mainly focuses on Collaborative filtering, Content-based Recommendation, and the Hybrid approach.

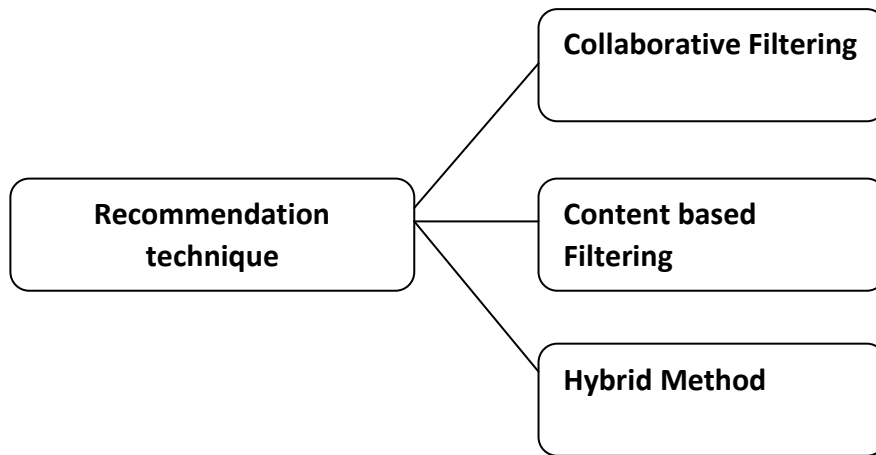


Fig.3.5: Recommender Techniques

3.6.2 COLLABORATIVE FILTERING (CF)

CF is considered to be the most popular and widely implemented technique in RS. Collaborative filtering recommends items based on similarity measures between:

1. Users
2. Items
3. Users and/or items

Items that are preferred by similar users are recommended to a user. It is determined by the similarity of the ratings of those items by the users who have rated both items. It focuses on the similarity of the user ratings for two items.

Collaborative filtering explores a technique for recommending items based on matching people with similar interests. CF is based on the assumption that similar users tend to like similar items. Three pillars of this approach are (1) many people must be engaged (so that the probability of a given person finding others with similar preferences will be high), (2) people representing their interests on the systems must be easy, and (3) algorithms must be able to match people with similar interests.

3.6.3 CONTENT-BASED RECOMMENDATION (CBR)

Content-based systems recommend items based on the properties of the item. For instance, if a user has watched many romantic movies then it recommends a movie categorized in the database as having the “romantic” genre. CBR focuses on attributes of the item. The similarity of items is determined by measuring similarities between their properties. It uses features of items determined by their similarity.

What must be done in a CBR System is to:

1. Construct for each item a profile which refers to item profiling
2. Construct a user profile

A profile is a record or collection of records representing important characteristics of the item. In simple cases, the profile consists of some characteristics of the item that are easily discovered e.g. consider the following features of a movie:

- Set of actors of the movie
- The director
- The year in which the movie was made
- The genre or general type of movie e.g. comedies, drama, romance.

The genre of movies is not readily available as part of the description of the movies. It is an ambiguous concept. Internet Movie Database (IMDB) assigns a genre/genres to every movie. The ultimate goal for CBR is to create both an item profile consisting of a feature–value pair and a user profile summarizing the preferences of the user based on their row in the utility matrix.

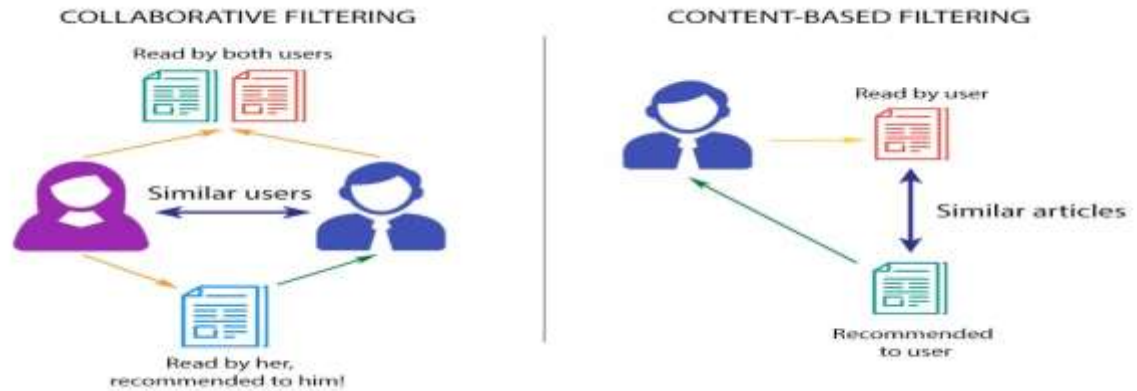


Fig.3.6: Overview of Recommendation Systems

3.6.3.1 THE STRENGTH AND WEAKNESS OF CONTENT-BASED RECOMMENDATION

When Content--based filtering is employed in RSs, it comes with several advantages compared to the Collaborative filtering approach:

1. **USER INDEPENDENCE** - Content-based recommenders rely completely upon on ratings that a given user provides to build her own profile. Instead, Collaborative filtering approach depends on ratings from other similar users in order to find the “nearest neighbors “of the given user. Similar users tend to have similar tastes since they rated the same items similarly. Then, the nearest neighbor’s preferences will be recommended to the given user.
2. **TRANSPARENCY** - CBR works by explicitly listing content features or descriptions that caused an item to occur in the list of recommendations. Those features are indicators to consult in order to decide on whether to trust a prediction accuracy of a recommendation. Contrarily, Collaborative systems are not as explicit as CBR since the only explanation for an item recommendation is that unknown users with similar tastes liked that item.
3. **NEW ITEM** - Content-based recommenders are capable of recommending items not yet rated by any user. Consequently, they do not suffer from the new-item problem, which affects Collaborative recommenders which depend solely on ratings from other similar users to make recommendations. Therefore, until the new item is rated by a considerable number of users, the system would not be able to recommend it.

Nonetheless, Content-based systems have several shortcomings:

1. **LIMITED CONTENT ANALYSIS** - Content-based techniques have a natural limit in the number and type of features that are associated with the objects they recommend. Domain knowledge is often needed, for example, in movie recommendations, the system needs to know the actors and directors, and sometimes, formal definition of entities and their relations are also needed. Content-based recommendation systems cannot provide suitable suggestions if the analyzed content does not contain adequate information to differentiate items the user likes, from items the user does not like. Some representations capture only certain aspects of the content, but there are many others that would influence a user's experience. For instance, often there is limited information in the word frequency to model the user's interests in jokes or poems, while techniques for effective computing would be most appropriate. Again, for web pages, feature extraction techniques from text completely ignore aesthetic qualities and additional multimedia. To sum it up, both automatic and manual assignment of features to items is not sufficient enough to define distinguishing aspects of items that turn out to be necessary for the elicitation of user interests.
2. **OVER-SPECIALIZATION** - Content-based recommenders have no inherent method for finding something unexpected. The system suggests items whose scores are high when matched against the user profile, hence the user is not thrilled with the recommended items because the items suggested are similar to those already rated. This drawback is also called the serendipity problem, highlighting the tendency of Content-based systems producing recommendations with a limited degree of novelty. As an example, when a user has only rated movies directed by Matt Damon, she will be recommended just those kind of movies. A "perfect" Content-based technique has difficulty in recommending anything new, limiting the range of applications for which it would be useful.

3. **NEW USER** - Enough ratings have to be collected before a Content-based recommender system can really understand user preferences and provide accurate recommendations. So, when few ratings are available, as for a new user, the system will not be able to provide reliable recommendations .

3.6.4 HYBRID RECOMMENDATION AND APPROACH

One common occurrence in RSs research is the demand to combine recommendation techniques to achieve peak performance. All of the known recommendation techniques have advantages and disadvantages, and many researchers have chosen to combine techniques in different ways in order to leverage their advantages. This session surveys the different hybrid recommendation approaches.

Hybrid systems combine two or more techniques in order to gain better performance with fewer limitations of each approach. Many hybrid systems have been applied to travel and tourism applications. For instance F. Ricci et al. illustrate a travel planning recommender system that is case-based, hence is knowledge-based, but also Collaborative-based since it recommends travel services that have been evaluated positively by others.

Fab is a recommendation system designed to help users explore the enormous amount of information available on the internet. This hybrid system combines the Content-based and Collaborative methods of recommendation in a way that exploits the advantages of the two approaches while avoiding their shortcomings. Fab's hybrid structure allows for automatic recognition of emergent issues relevant to various groups of users. It also enables two scaling problems pertaining to the rising number of users and documents, to be addressed.

One major tactic for improving recommendation is to combine Collaborative filtering with Content-based recommenders. We can illustrate the benefits of such hybrid systems with a simple example; suppose one user has rated the NBA page

from CBSSports.com favorably, while another has rated the NBA page from CNN.com favorably, pure Collaborative filtering would find no correlation between the two users. However, Content analysis can show that the two items are in fact quite similar, thus indicating a match between the users. The Fab system builds on this intuition. It analyzes the content of items that users rate favorably to build Content-based profiles of user interest. It then applies Collaborative filtering techniques to identify other users with similar interests. In another effort, the Group Lens research group is testing by using Collaborative filtering as a technique to combine the opinions of other users and personal information filtering agents.

3.7 DATA ANALYSIS

3.7.1 Data Collection

To collect movie data here we are using Web API and web scrapping.

API

An **application programming interface (API)** is a connection between computers or between computer programs. It is a type of software interface, offering a service to other pieces of software.^[1] A document or standard that describes how to build such a connection or interface is called an *API specification*. A computer system that meets this standard is said to *implement* or *expose* an API. The term API may refer either to the specification or to the implementation.

Web-Scrapping

Web scraping is the process of using bots to extract content and data from a website. Unlike screen scraping, which only copies pixels displayed onscreen, web scraping extracts underlying HTML code and, with it, data stored in a database. The scraper can then replicate entire website content elsewhere. Web scraping is used in a variety of digital businesses that rely on data harvesting. Legitimate use cases include:

- Search engine bots crawling a site, analysing its content and then ranking it.

- Price comparison sites deploying bots to auto-fetch prices and product descriptions for allied seller websites.
- Market research companies using scrapers to pull data from forums and social media (e.g., for sentiment analysis).

TMDB API

TMDB is a site that provides a free API portal for researchers who are interested in getting access to movie data. It is free to register for an account and get a free personal API key (an API key is a crucial piece of information for accessing any API portal, think about it like a digital ticket for an event).

Movie Collection:

For collecting the data of Bollywood movies we are using python modules such as:

- requests: Requests library is an integral part of Python for making HTTP requests to a specified URL. Whether it be REST APIs or Web Scrapping, requests is must to be learned for proceeding further with these technologies. When one makes a request to a URI, it returns a response.
- html5lib: A pure-python library for parsing HTML. It is designed to conform to the WHATWG HTML specification, as is implemented by all major web browsers.
- bs4: BeautifulSoup object is provided by Beautiful Soup which is a web scraping framework for Python. Web scraping is the process of extracting data from the website using automated tools to make the process faster.

And for collecting the data of Hollywood movies we are using the TMDB API:

How to process the API data:

- Before accessing an API, the first step is to read the API documentation; this step is crucial
- because in order to know what data to pull, how to pull it and how to unpack it.
- We need to use the “get movie detail” query to pull the movie detail for each movie.
- By substituting “API key” and providing a movie id of interest, we can form an

API query.

- JSON is the format most API portals return their data in, so we need a way to extract the data

we need from this jiberish and put it in a format we can easily digest as humans (say CSV).

- All JSON returns from API portals can be treated as a dictionary when used in Python.
- Finally, start by importing all the packages that we need.
- Next, use the *requests* package to query the API; this step will return the result in JSON format.
- The last step is to unpack the result and write it to a CSV file.

3.7.2 Data Pre-Processing

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

- After getting our initial data from web scraping, where we have used Wikipedia
- The data here is further pre - processed using panda and numpy.

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

Numpy: NumPy stands for Numerical Python. NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices.

How the data is processed:

- The data in the system has to be stored and retrieved from the database.
- Data elements and data structures to be stored have been identified at the analysis stage. They are structured and put together to design the data storage and retrieval system.

Normalization is done to get an internal consistency of data and to have minimum redundancy and maximum stability. This ensures minimizing data storage required, minimizing chances of data inconsistencies, and optimizing for updates.

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

3.8.1 Cosine Similarity: It 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:

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

3.8.2 KNN:

For item based collaborative filtering implementation, KNN is perfect model and it has good baseline for development of recommendation system . But what is the KNN? KNN is a non-parametric, lazy learning method. KNN use database where data points were separated in to many clusters to make inference for new samples.KNN depends on item feature similarity and will not make assumptions on underlying distributed data . While KNN making inference over movie, it calculates distance between target movie and every other movie which is in the database, after it ranks based on distance between them and returns top K nearest neighbor movies as the most similar movie recommendations.

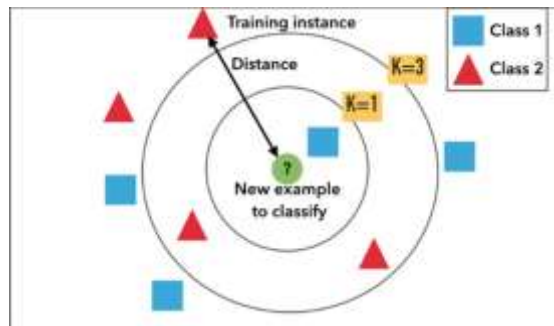


Fig.3.7. Classification about new sample.

3.8.3 Svd

The Singular Value Decomposition is a method from linear algebra that is used as a dimensionality reduction technique in machine learning. SVD is an unsupervised machine learning algorithm. SVD is a matrix factorization technique, that reduces the number of features of a dataset by reducing the space dimension from N-dimension to K-dimension (where K is less than N). SVD is used in both matrix operations such as inverse and dimensionality reduction. SVD can also be used in denoising data, least squares linear regression and image compression. SVD is calculated as follows:

$$A = U \cdot \text{Sigma} \cdot V^T$$

where A is mxn matrix that we decompose, U is mxm matrix, sigma is mxn diagonal matrix and V^T is transpose of nxn matrix. In movie recommender systems, SVD is a collaborative filtering technique.

Applications:

This is by no means an exhaustive list of SVD's application.

Dimensionality Reduction:

Due to its relationships with PCA, we can imagine a very frequent use of SVD is feature reduction. By only selecting the top singular values, we have in effect compressed the original information and represented it using fewer features. Note that because SVD is also a numerical algorithm, it is important to standardize the features to ensure the magnitude of the entries are of similar range.

Information Retrieval:

SVD has also been used extensively in information retrieval, in this particular application, it is also known as Latent Semantic Analysis (LSA) or Latent Semantic Indexing (LSI). As we'll soon see, the idea is very similar to topic modeling. The fundamental problem in information retrieval is: given some search terms, retrieve all of the documents that contain those search terms or, perhaps more usefully, return documents whose content is semantically related to the search terms. For example, if one of the search terms was "automobile" it might be appropriate to return also documents that contain the search term "car".

CHAPTER 4

RESULT AND DISCUSSION

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

In our project we have developed a hybrid approach i.e combination of both content and collaborative filtering .Both the approaches have advantages and disadvantages.

In **content based filtering** it is based on the user ratings or user likes only such kind of movie will recommended to the user. **Advantages:** it is easy to design and it takes less time to compute. **Dis-advantages:** the model can only make recommendations based on existing. Interests of the user. In other words, the model has limited ability to expand on the users' existing interests.

In **Collaborative filtering** the recommendation is comparison of similar users. **Advantages:** No need domain knowledge because the embeddings are automatically learned. The model can help users discover new interests. In isolation, the ML system may not know the user is interested in a given item, but the model might still recommend it because similar users are interested in that item.

Dis-advantages: The prediction of the model for a given (user, item) pair is the dot product of the corresponding embeddings. So, if an item is not seen during training, the system can't create an embedding for it and can't query the model with this item. This issue is often called the cold-start problem.

The **hybrid approach** will resolves all these limitations by combining both content and collaborative filtering. The main **disadvantage** in hybrid approach is it require high memory.

Table.4.1: Comparison between the three approaches

PARAMETERS	COLLABORATIVE APPROACH	CONTENT BASED APPROACH	HYBRID APPROACH
Accuracy	Low	Average	High
Quality	Low	Average	High
Scalability	Less	Average	High
Computing Time	Average	High	Low
Memory	Average	Low	High

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 CONCLUSION

In this project, to improve the accuracy, quality and scalability of movie recommendation system, a Hybrid approach by unifying content based filtering and collaborative filtering; using Singular Value Decomposition (SVD) as a classifier and Cosine Similarity is presented in the proposed methodology. Existing pure approaches and proposed hybrid approach is implemented on three different Movie datasets and the results are compared among them. Comparative results depicts that the proposed approach shows an improvement in the accuracy, quality and scalability of the movie recommendation system than the pure approaches. Also, computing time of the proposed approach is lesser than the other two pure approaches.

5.2 FUTURE SCOPE

In the proposed approach, It has considered Genres of movies but, in future we can also consider age of user as according to the age movie preferences also changes, like for example, during our childhood we like animated movies more as compared to other movies. There is a need to work on the memory requirements of the proposed approach in the future. The proposed approach has been implemented here on different movie datasets only. It can also be implemented on the Film Affinity and Netflix datasets and the performance can be computed in the future.

REFERENCES

- [1] Anmol Chauhan , Deepank Nagar , Prashant Chaudhary (2021). "Movie Recommender system using sentiment analysis" International conference of innovative practices in technology and management ICIPTM52218.2021.9388340
- [2] Lavanya, R., Tanmay Khokle, Abhideep Maity. (2020). "Review on Hybrid Recommender System for Mobile Devices" Springer Lecture notes on networks and systems, 130, 477-486.
- [3] Liaoliang Jiang, Yuting Cheng, L. Y. J. L. H. Y. X. W. (2019). "A trust based collaborative filtering algorithm for e commerce recommendation system." Journal of Ambient Intelligence and Humanized Computing, 10, 3023–3034.
- [4] Lavanya, R., Rithika Lahari, PalakGupta. (2019). "An Optimal Enhancement of the Dynamic Features of Recommender Systems" International Journal of Recent Technology and Engineering, 8(2S4), 51-55.
- [5] Yi Ren, Jingke Xu, Jie Huang and Cuirong Chi " Research on Collaborative Filtering Recommendation Algorithm for Personalized Recommendation System", 2019 9th International Conference on Education and Social Science (ICESSE 2019).
- [6] Vishwa, Bhavesh, Aman Gupta, Pranal Soni "Movie Recommendation System", International Research Journal of Engg and Technology Volume: 05 Issue: 02, Feb-2018
- [7] S. G Walunj, K Sadafale, "An online recommendation system for e-commerce based on Apache Mahout framework", 2017 ACM SIGMIS International Conference on Computers and People Research, pp. 153- 158,2013.
- [8] H. W. Chen, Y. L. Wu, M. K. Hor and C. Y. Tang, "Fully content-based movie recommender system with feature extraction using neural network," 2017 International Conference on Machine Learning and Cybernetics (ICMLC), Ningbo, China, 2017, pp. 504-509. doi: 10.1109/ICMLC.2017.8108968.

APPENDICES

A.SOURCE CODE

```
import pandas as pd
import numpy as np
import re
from sklearn.metrics.pairwise import cosine_similarity
from nltk.corpus import stopwords
stopwords_english = set(stopwords.words('english'))
from nltk.stem import PorterStemmer
stemmer = PorterStemmer()
from ast import literal_eval

import warnings
warnings.filterwarnings("ignore")

data = pd.read_csv('datasets/movies_metadata.csv', low_memory =
False)[['id','original_title', 'imdb_id', 'genres', 'overview', 'vote_average',
'vote_count']].dropna().set_index('id')
credits = pd.read_csv('datasets/credits.csv', low_memory =
False).set_index('id').dropna()

#Cleaning Genres
def split_genres(row):
    row['genres'] = " ".join([info['name'] for info in literal_eval(row['genres'])])
    return row
data = data.apply(split_genres, axis=1)

#finding director
def find_director(row):
    try:
        for info in literal_eval(row['crew']):
```

```

        if info['job'] == 'Director':
            row['director'] = info['name']
            break
    except:
        row['director'] = pd.NA
    return row
credits = credits.apply(find_director, axis=1).drop(['cast', 'crew'], axis=1)
credits.index = credits.index.astype(str)

#merging two datasets
data = pd.merge(data, credits, left_index = True, right_index = True, how = 'left')
data.index = data.index.astype(int)

#function to generate top movies
def top_movies(num = 100, quantile = 0.95, data = data):
    req_votes = np.quantile(data['vote_count'], quantile)

    new_data = data[data['vote_count']>req_votes]
    means = new_data['vote_average']
    tot_mean = new_data['vote_average'].mean()
    votes = new_data['vote_count']

    new_data['ratings'] = (votes/(votes+req_votes)*means +
    req_votes/(votes+req_votes)*tot_mean)

    return new_data.sort_values('ratings', ascending =
False).iloc[:num,:].reset_index(drop=True)

#Function to clean the text and create Word-Document-Frequency/Title-Body Split
def clean(x):
    res = []

```

```

for w in x:
    if w not in stopwords_english:
        w = stemmer.stem(re.sub('[^a-zA-z0-9]', '', w))
        if(len(w)>0):
            res.append(w)
return res

def preprocess(data):
    stemmer = PorterStemmer()
    docs = []
    DF = {}

    for ind, row in data.iterrows():
        set_title = set()
        doc_count = 0
        title = row['original_title'].lower().strip().split()
        title += row['genres'].lower().strip().split()
        title += [row['director'].lower().strip()]
        body = row['overview'].lower().strip().split()
        body = clean(body)

        for word in title+body:
            if word in DF:
                if row.name in DF[word]:
                    DF[word][row.name] += 1
                else:
                    DF[word][row.name] = 1
            else:
                DF[word] = {}
                DF[word][row.name] = 1

```

```

        doc_count+=1
        if word in title:
            set_title.add(word)
        docs.append((doc_count, set_title))

    return DF, docs

def generate_cosine_tfidf(data, alpha = 0.6):
    DF, docs = preprocess(data)
    tf_idf = np.zeros((len(docs), len(DF)))

    for i, (word, key) in enumerate(DF.items()):
        w_count = np.sum([value for _, value in key.items()])
        for ind, value in key.items():
            tf = value/docs[ind][0]
            idf = np.log(len(docs)/(w_count+1))
            tfidf = tf*idf*(1-alpha)
            if word in docs[ind][1]:
                tfidf = tf*idf*alpha

        tf_idf[ind][i] = tfidf
    return cosine_similarity(tf_idf)

movies = top_movies(500)
tf_idf = generate_cosine_tfidf(movies, alpha = 0.4)

#Function to predict the top movies given a choice
def predict_movies(movie_name = "The Dark Knight", num = 10, verbose = 0, out =
True, data = movies, tf_idf = tf_idf):
    try:
        ind = data[data['original_title'] == movie_name].index[0]

```

```

output = data.loc[[val for val in np.argsort(tf_idf[ind])[:-1][1:num+1]]]
if verbose == 2:
    print("The TF-IDF Cosine similarity scores for relevant movies are as
follows:\n")
    print([round(val,2) for val in np.sort(tf_idf[ind])[:-1][1:num+1]], '\n')
if verbose >= 1:
    print(f'The top {num} recommended movies for "{movie_name}" are as
follows:\n')
    for ind, row in output.iterrows():
        print("Title: {}".format(row['original_title']))
        print("Rating: {}".format(round(row['ratings'], 1)))
        print("Genres: {}".format(row['genres']))
        print("IMDB Link: https://www.imdb.com/title/{}".format(row['imdb_id']))
        print("*****\n")
    if(out):
        return output
except:
    print("MOVIE NOT FOUND!")

```

#Generate the top 2000 movie chart and calculate the Cosine Similarity Matrix

```
movies = top_movies(2000)
```

```
tf_idf = generate_cosine_tfidf(movies, alpha = 0.6)
```

```
predict_movies("The Dark Knight", num = 10, verbose = 2, out = False, data =
movies, tf_idf = tf_idf)
```

```
predict_movies("The Hobbit: An Unexpected Journey", num = 10, verbose = 2, out =
False, data = movies, tf_idf = tf_idf)
```

B.SCREENSHOTS

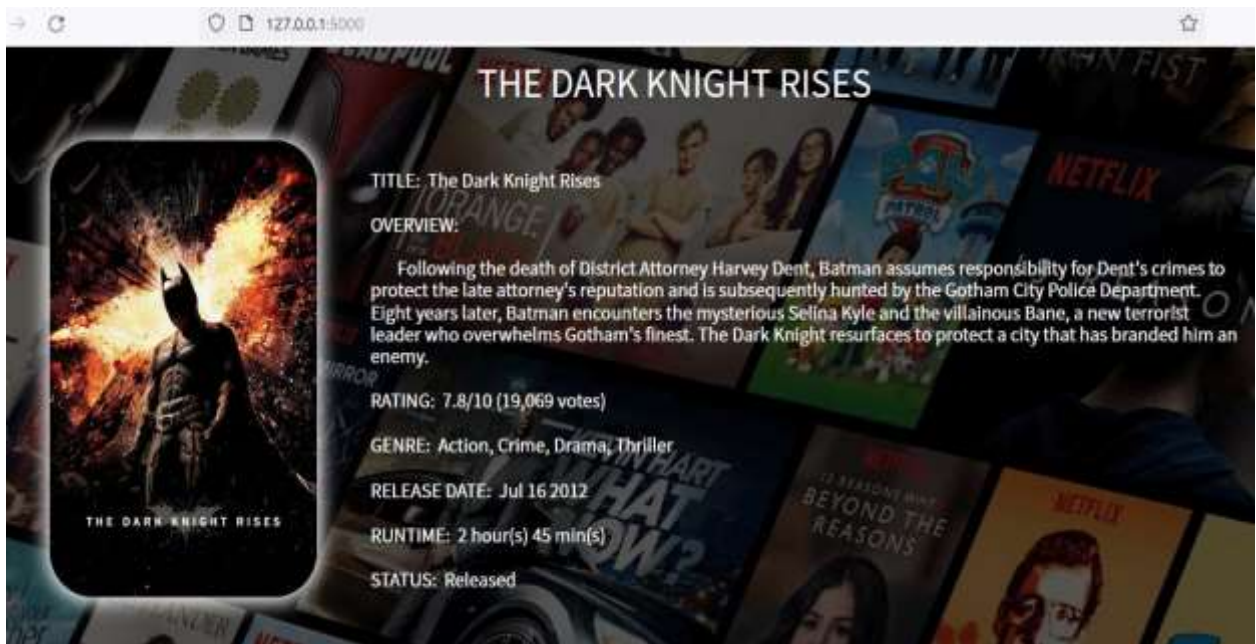


Fig. Home page

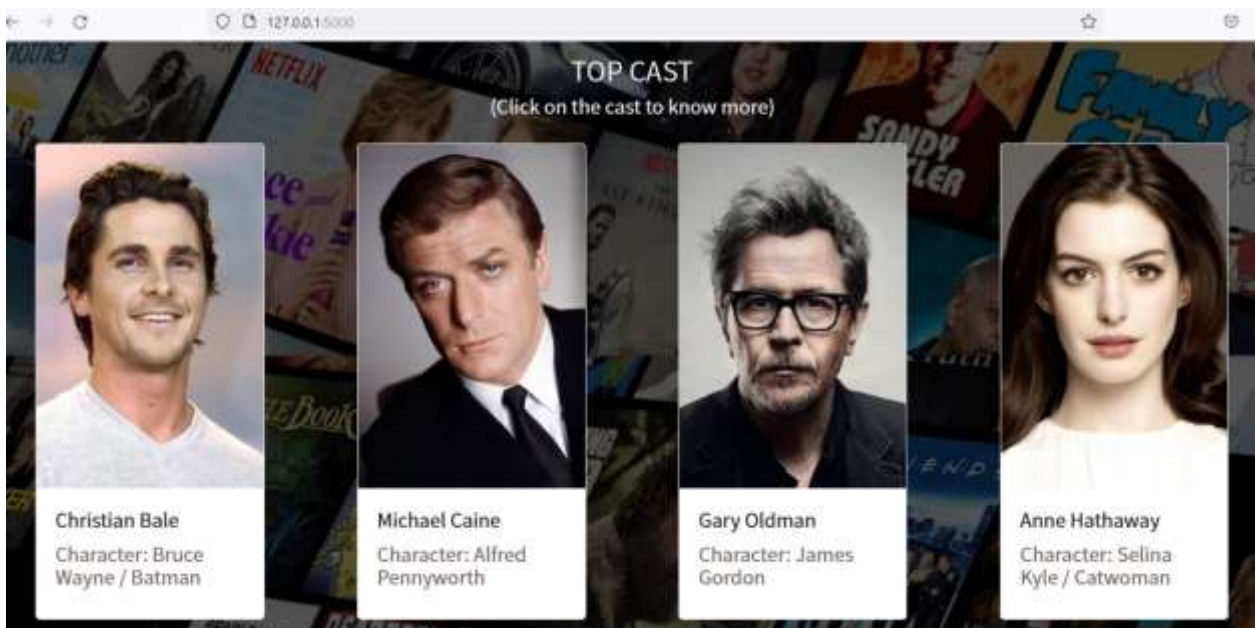


Fig. Top cast in movie

USER REVIEWS	
Comments	Sentiments
I am not a huge comic book-based movie fan, though I see most all of them. That's because my daughter is nuts for comic book movies and we always seem to see them. Some I have loved ("Scott Pilgrim", "The Dark Knight" and "Captain America" come to mind) and some I didn't ("Daredevil")--but mostly I do enjoy them. While I am sure I am bucking the tide here, I was NOT especially thrilled with "The Dark Knight Rises". This doesn't mean I disliked it--it just left me a bit flat. Now the problem was NOT the movie's length--I like longer films and my butt didn't fall asleep during the film (which is a very good sign)! My problems were more about the incredibly dark tone of the film. Unlike "The Dark Knight", "The Dark Knight Rises" really lacked fun and was 100% devoid of any humor to break up the dark mood. In fact, I kept hoping Tony Stark ("Ironman") would do a crossover from the Marvel films and lighten the mood! Stark (Robert Downey) managed to do a Bruce Wayne-like character but infuse him with energy and likability--Christian Bale just seemed tired and limp and humorless. This was more obvious in this film as opposed to the last because there was no Joker-like villain to provide comic relief (albeit darkly comic). The closest they got was Catwoman (Anne Hathaway)--who was a GREAT character but who wasn't really necessary for the plot. As for Bane and his friends, they were scary and nasty...and nothing more. When the film was over, I realized I had a reasonably good time but nothing more. If some other director and actor pick up the franchise, my advise is work more on the characters--make them more likable and interesting--even the villains.	Good : 😊
Possibly my personal favourite of the Nolan Batman films, and I do like Batman Begins and The Dark Knight very much. Is it perfect? For me, not quite, there are so many plot strand in a long running time that the film occasionally (and there is a big emphasis on that) sags in momentum and some of the strands in the ending could have done with more development. It is a brilliant film all the same, one of the highlights so far of the year. As to expect from Christopher Nolan, it is a visually audacious film, with gorgeous cinematography, a beautifully realised Gotham City and great special effects. Hans Zimmer's score is not one	Good : 😊
Possibly my personal favourite of the Nolan Batman films, and I do like Batman Begins and The Dark Knight very much. Is it perfect? For me, not quite, there are so many plot strand in a long running time that the film occasionally (and there is a big emphasis on that) sags in momentum and some of the strands in the ending could have done with more development. It is a brilliant film all the same, one of the highlights so far of the year. As to expect from Christopher Nolan, it is a visually audacious film, with gorgeous cinematography, a beautifully realised Gotham City and great special effects. Hans Zimmer's score is not one of his very best for me, but there is still an epic and haunting quality that is just perfect for this film. The dialogue provokes thought and makes an emotional impact, and a vast majority of the time The Dark Knight Rises is ambitious and gripping. I don't consider Christopher Nolan one of my favourites, this said he is a hugely impressive director who shows great ambition and a willingness to make his subject matters interesting. He does a really wonderful job here, as the set/action pieces including the plane hijack and Bane's assault on Gotham are enough to make the jaw drop. The performances are just terrific from all involved, Matthew Modine fares weakest but he is far from bad. Christian Bale is still a charismatic presence, Matthew Caine appropriately loyal and Gary Oldman and Morgan Freeman subtly understated. Joseph Gordon Levitt captures his character's idealism wonderfully. Tom Hardy is in beefed-up terrifying mode as Bane, a worthy successor to Heath Ledger's iconic Joker. Anne Hathaway was a curious casting choice, but as Catwoman she is spot on, here is a Catwoman who actually does much more than acting sexy (which she still is), she is very ruthless and no-nonsense here. All in all, a hugely impressive film, the best of Nolan's Batman trilogy- though I'm prepared for any flaming I'll get for this opinion- and one of the highlights so far of 2012. 9/10 Bethany Cox	Good : 😊
the great virtue of Christopher Nolan remains to propose the only Batman realistic version. not the best. only the only authentic. this film is the precise pledge because it is not only about a hero or a town. but about the past, force of memories, subtle cruelty of revenge, the right end for a magnificent series and about the right gestures for build a new beginning. the meet of a great cast and the best script. this is all. and the argument for define "The Dark Knight Rises" as a masterpiece for many reasons, not only artistic, but, first, for to be /becomes a story inside you. many scenes are familiar. from books or other films, from stories or paintings, from nightmares or every day reality. and the genius of Nolan is not to discover them. but use them. in the best manner. an impeccable film. this is the word for define it.	Good : 😊

Fig. reviews for the movie

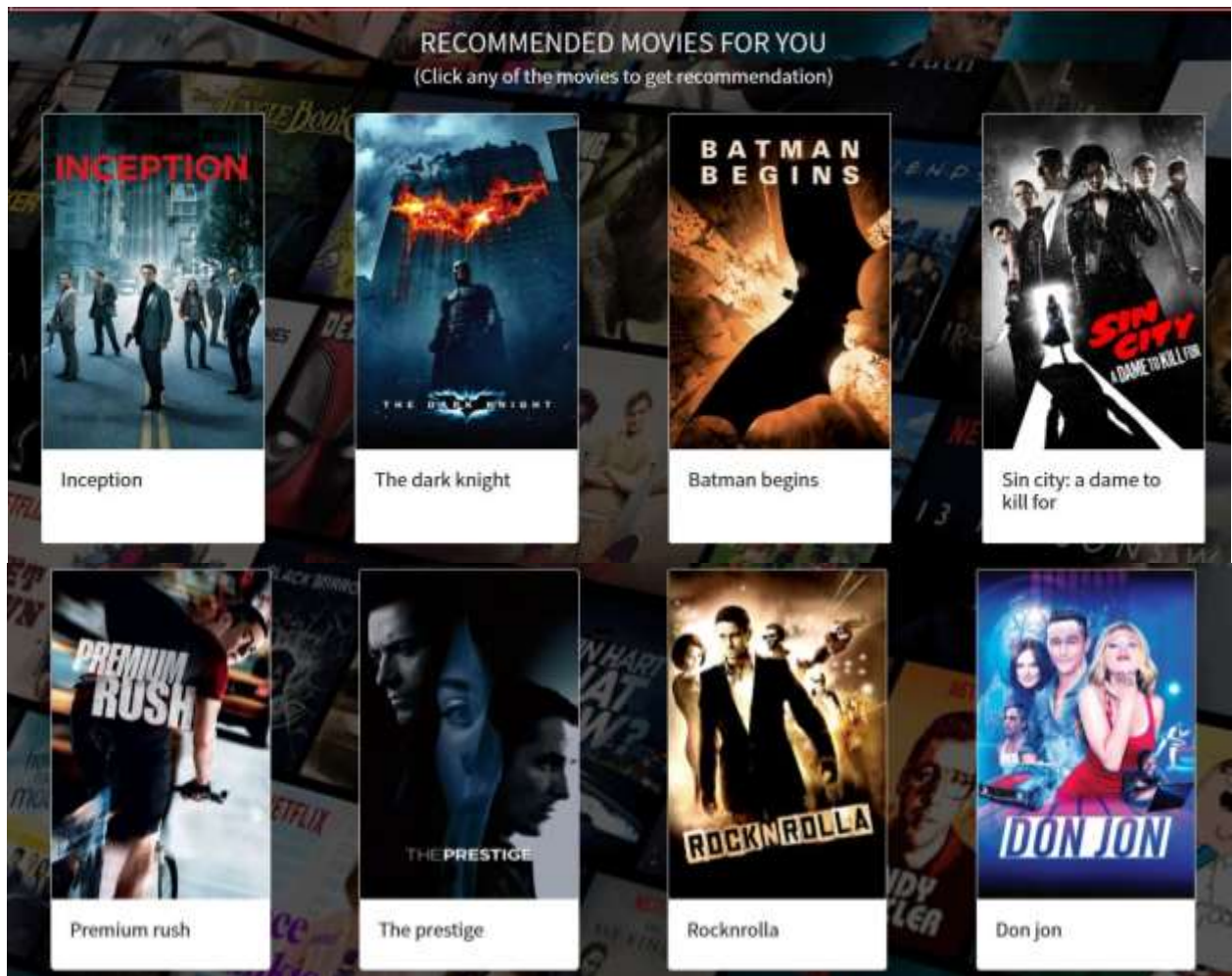


fig.recommended movie

C.PUBLICATION WITH PLAGARISM REPORT

- submitted in NCCICN 2022(National Conference on computational Intelligence and Communication Networks)

