

# Movie Recommendation System

## Bachelor of Technology Computer Science & Engineering

by

Muskan (191020101008)  
Sandeep Sinha (191020101009)



COMPUTER SCIENCE & ENGINEERING DEPARTMENT OM  
STERLING GLOBAL UNIVERSITY HISAR,  
125001 INDIA

May, 2023

# Movie Recommendation System

*A Project Report submitted*  
in partial fulfillment of the requirements  
for the Degree of

Bachelor of Technology in Computer Science &  
Engineering

by

Muskan (191020101008)  
Sandeep Sinha (191020101009)

*Under the guidance of*

Dr. Parveen Sehgal  
Professor  
Dept. Of Computer Science & Engineering



to the

COMPUTER SCIENCE & ENGINEERING DEPARTMENT  
OM STERLING GLOBAL UNIVERSITY  
HISAR, 125001 INDIA

May, 2023

# **CERTIFICATE**

This is to certify that the project work entitled “**MOVIE RECOMMENDATION SYSTEM**” carried out by **Ms. Muskan and Mr. Sandeep Sinha** , Bonafide students of **Om Sterling Global University** in partial fulfillment for the award of degree in **Bachelor of Technology in Computer Science and Engineering** during the year 2022-2023 .It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library.

The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

**Date**

**Dr. Parveen Sehgal**  
**Professor**

**Dept of Computer Science & Engineering**

# DECLARATION

We, the students of Computer Science and Engineering, Om Sterling Global University Hisar declare that the work entitled "**MOVIE RECOMMENDATION SYSTEM**" has been successfully completed under the guidance of **Prof. Dr. Parveen Sehgal, Computer Science and Engineering Department, Om Sterling Global University Hisar**. This dissertation work is submitted in partial fulfillment of the requirements for the award of Degree of Bachelor of Engineering in Computer Science and Engineering during the academic year 2019 - 2023. Further the matter embodied in the project report has not been submitted previously by anybody for the award of any degree or diploma to any university.

Place:

Date:

Team members:

Muskan

Btech CSE (8<sup>th</sup> Sem)191020101008

Sandeep Sinha

Btech CSE (8<sup>th</sup> Sem)191020101009

# ACKNOWLEDGEMENT

We take this opportunity to express our sincere gratitude and respect to Om Sterling Global University for providing us a platform to pursue my studies and carry out our final year project we would like to thank **Dr. Parveen Sehgal** Professor of Department of Computer Science and Engineering, OSGU Hisar who has been a constant Support and encouragement throughout the course of this project.

We have a great pleasure in expressing our deep sense of gratitude to **Dr. Rajinder Singh Sodhi, Head of Department of Computer Science & Engineering** , for his constant encouragement. We also extend my thanks to all the faculty of Computer Science and Engineering who directly or indirectly encouraged us.

We would also like to thank our classmates for sustaining and guiding me to carry out our project work successfully. And last, but not least we would like to thank our parents without thier support this project couldn't be completed.

Muskan

Sandeep Sinha

## **ABSTRACT**

In this hustling world, entertainment is a necessity for each one of us to refresh our mood and energy. Entertainment regains our confidence for work and we can work more enthusiastically.

For revitalizing ourselves, we can listen to our preferred music or can watch movies of our choice. For watching favourable movies online we can utilize movie recommendation systems, which are more reliable, since searching of preferred movies will require more and more time which one cannot afford to waste. In this paper, to improve the quality of a movie recommendation system, a Hybrid approach by combining content based filtering and collaborative filtering, using Support Vector Machine as a classifier and genetic algorithm is presented in the proposed methodology and comparative results have been shown which depicts that the proposed approach shows an improvement in the accuracy, quality and scalability of the movie recommendation system than the pure approaches in three different datasets. Hybrid approach helps to get the advantages from both the approaches as well as tries to eliminate the drawbacks of h methods

# TABLE OF CONTENTS

<b>ABSTRACT.....</b>	<b>5</b>
<b>Introduction.....</b>	<b>8</b>
1.1    Relevance of the Project.....	8
1.2    Problem Statement.....	9
1.3    Objective of the Projects.....	9
1.4    Scope of the Project.....	9
1.5    Methodology for Movie Recommendation.....	10
1.6    Agile Methodology.....	11
<b>LITERATURE SURVEY.....</b>	<b>12</b>
2.1    Movie Recommendation System by K-MeansClustering AND K-Nearest Neighbour.....	12
2.2    Movie Recommendation system Using Collaborative Filtering.....	13
<b>SYSTEM REQUIREMENTS SPECIFICATION.....</b>	<b>14</b>
3.1    Hardware Requirements.....	14
3.2    Software Specification.....	14
3.3    Software Requirements.....	14
<b>SYSTEM ANALYSIS AND DESIGN.....</b>	<b>16</b>
4.1    System Architecture of Proposed System.....	16
4.2    Activity Diagram.....	17
4.3    Dataflow.....	18
<b>IMPLEMENTATION.....</b>	<b>19</b>
5.2    Singular Value Decomposition (SVD):.....	19
Experimental requirementsCode: Jupyter notebook.....	20
<b>RESULTS AND DISCUSSION.....</b>	<b>22</b>
<b>TESTING.....</b>	<b>24</b>
7.1 Testing Methodologies.....	24
<b>CONCLUSION AND FUTURE SCOPE.....</b>	<b>26</b>
8.2    Future scope.....	26
<b>REFERENCES.....</b>	<b>27</b>

# LIST OF FIGURES

FIGURES NAME	Page No.
Fig 4.1 Architecture for hybrid approach	8
Fig 4.2 Activity diagram	9
Fig 4.3 Data Flow	10
Fig 5.1 Code snippet	12
Fig 5.2 Backed code snippet	13
Fig 6.1 User login	15
Fig 6.2 List of recommended movies	15



# 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 is Improving the Scalability.

Enhancing the user experience.

## 1.4 Scope of the Project

**1.Data collection:** The first step in building a movie recommendation system is to collect data on movies and users. This can include information such as movie titles, ratings, genres, cast, crew, and user demographics.

**2.Data preprocessing:** Once the data has been collected, it needs to be cleaned and processed to prepare it for use in the recommendation system. This can involve tasks such as removing duplicates, filling in missing values, and encoding categorical variables.

**3.Feature engineering:** Feature engineering involves selecting and transforming the most relevant features from the data to be used in the recommendation algorithm. This can include features such as user preferences, movie popularity, and similarity between movies.

**4.Recommendation algorithm:** The recommendation algorithm is the heart of the system and is responsible for generating recommendations for users. Common algorithms include collaborative filtering, content-based filtering, and hybrid approaches.

**5.User interface:** The user interface is the component that allows users to interact with the recommendation system. This can include features such as search functionality, user profiles, and personalized recommendations.

**6. Evaluation:** Evaluation involves measuring the effectiveness of the recommendation

system. This can include metrics such as precision, recall, and coverage.

## **1.5 Methodology for Movie Recommendation**

The hybrid approach proposed an integrative method by merging fuzzy k- means clustering method and genetic algorithm based weighted similarity measure to construct a movie recommendation system. The proposed movie recommendation system gives finer similarity metrics and quality than the existing Movie recommendation system but the computation time which is taken by the proposed recommendation system is more than the existing recommendation system. This problem can be fixed by taking the clustered data points as an input dataset

The proposed approach is for improving the scalability and quality of the movie recommendation system .We use a Hybrid approach , by unifying Content-Based Filtering and Collaborative Filtering, so that the approaches can be profited from each other. For computing similarity between the different movies in the given dataset efficiently and in leasttime and to reduce computation time of the movie recommender engine we used cosine similarity measure.

## 1.6 Agile Methodology

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

**2. Data Analysis:** make sure that that the collected data sets are correct and analysing the data in the csv files. i.e. checking whether all the column Felds are present in the data sets.

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

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

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

# LITERATURE SURVEY

Over the years, many recommendation systems have been developed using either collaborative, content based or hybrid filtering methods. These systems have been implemented using various big data and machine learning algorithms.

## **2.1 Movie Recommendation System by K-Means Clustering AND K-Nearest Neighbour**

A recommendation system collect data about the user's preferences either implicitly or explicitly on different items like movies. An implicit acquisition in the development of movie recommendation system uses the user's behaviour while watching the movies. On the other hand, a explicit acquisition in the development of movie recommendation system uses the user's previous ratings or history. The other supporting technique that are used in the development of recommendation system is clustering. Clustering is a process to group a set of objects in such a way that objects in the same clusters are more similar to each other than to those in other clusters. K- Means Clustering along with K-Nearest Neighbour is implemented on the movie lens dataset in order to obtain the best-optimized result. In existing technique, the data is scattered which results in a high number of clusters while in the proposed technique data is gathered and results in a low number of clusters.

The process of recommendation of a movie is optimized in the proposed scheme. The proposed recommender system predicts the user's preference of a movie on the basis of different parameters. The recommender system works on the concept that people are having common preference or choice. These users will influence on each other's opinions. This process optimizes the process and having lower RMSE.

## **2.2 Movie Recommendation system Using Collaborative Filtering**

By Ching-Seh (Mike) Wu, Deepti Garg, Unnathi Bhandary

Collaborative filtering systems analyse the user's behaviour and preferences and predict what they would like based on similarity with other users. There are two kinds of collaborative filtering systems; user-based recommender and item-based recommender.

### **2.2.1 Use-based filtering:**

User-based preferences are very common in the field of designing personalized systems. This approach is based on the user's likings. The process starts with users giving ratings (1-5) to some movies.

These ratings can be implicit or explicit. Explicit ratings are when the user explicitly rates the item on some scale or indicates a thumbs-up/thumbs-down to the item. Often explicit ratings are hard to gather as not every user is much interested in providing feedbacks. In these scenarios, we gather implicit ratings based on their behaviour. For instance, if a user buys a product more than once, it indicates a positive preference. In context to movie systems, we can imply that if a user watches the entire movie, he/she has some likeability to it. Note that there are no clear rules in determining implicit ratings. Next, for each user, we first find some defined number of nearest neighbours. We calculate correlation between users' ratings using Pearson Correlation algorithm. The assumption that if two users' ratings are highly correlated, then these two users must enjoy similar items and products is used to recommend items to users.

### **2.2.2 Item-based filtering:**

Unlike the user-based filtering method, item-based focuses on the similarity between the item's users like instead of the users themselves. The most similar items are computed ahead of time. Then

for recommendation, the items that are most similar to the target item are recommended to the user.

# SYSTEM REQUIREMENTS SPECIFICATION

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

## 3.1 Hardware Requirements

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

## 3.2 Software Specification

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

## 3.3 Software Requirements

### 3.3.1 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.<sup>3</sup>

### 3.3.3 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.

**Sk learn :** 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



# SYSTEM ANALYSIS AND DESIGN

## 4.1 System Architecture of Proposed System

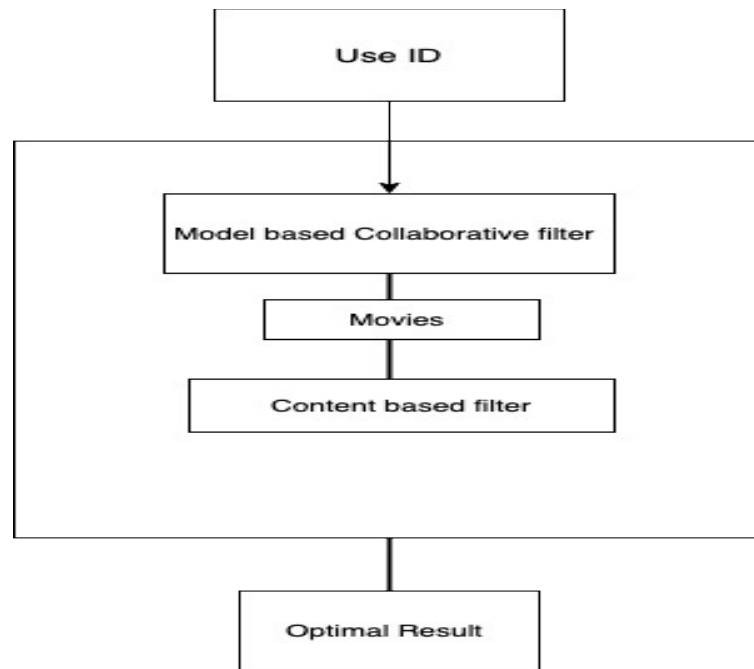


Fig:-4.1 Architecture for hybrid approach

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.

## 4.2 Activity Diagram

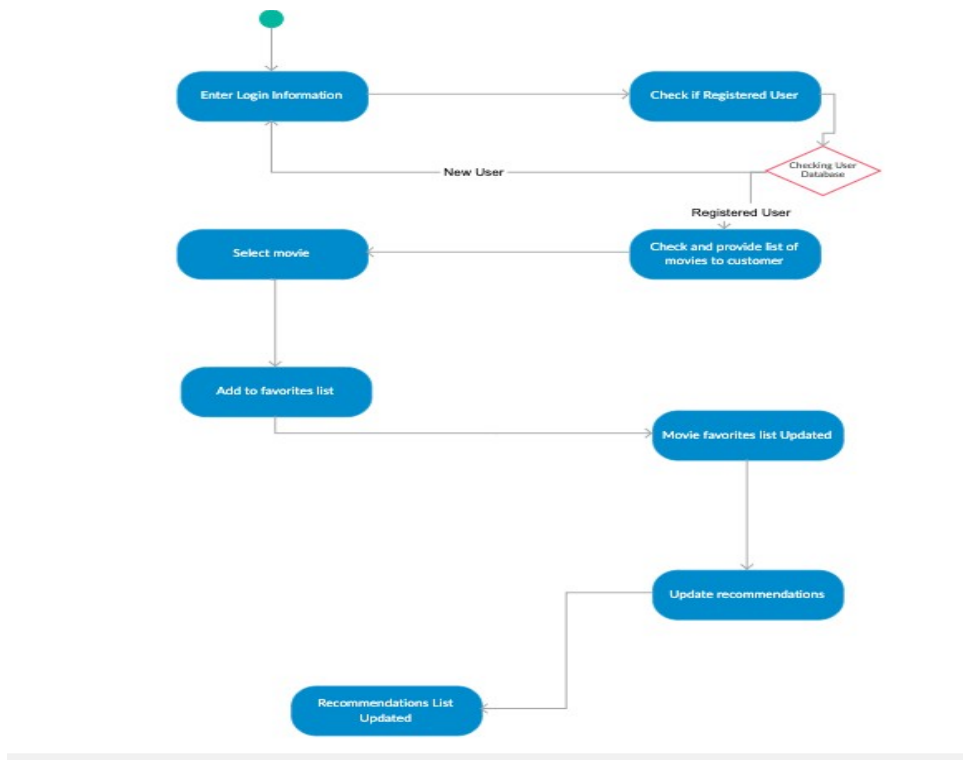


Fig:-4.2 Activity diagram

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

## 4.3 Data flow

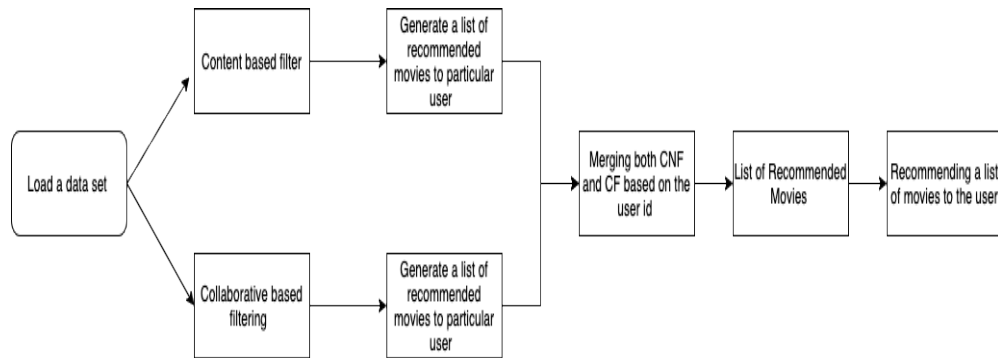


Fig:-4.3 Data Flow Diagram

Initially load the data sets that are required to build a model the data set that are required in this project are movies.csv, ratinf.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 useid a single final list of movies are recommended to the particular user

# IMPLEMENTATION

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

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

Formula:

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

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

## 5.2 Singular Value Decomposition (SVD):

Let A be an n\*d matrix with singular vectors  $v_1, v_2, \dots, v_r$  and corresponding singular values  $\sigma_1, \sigma_2, \dots, \sigma_r$ . Then  $u_i = (1/\sigma_i) A v_i$ , for  $i = 1, 2, \dots, r$ , are the left singular vectors and by Theorem 1.5, A can be decomposed into a sum of rank one matrices a

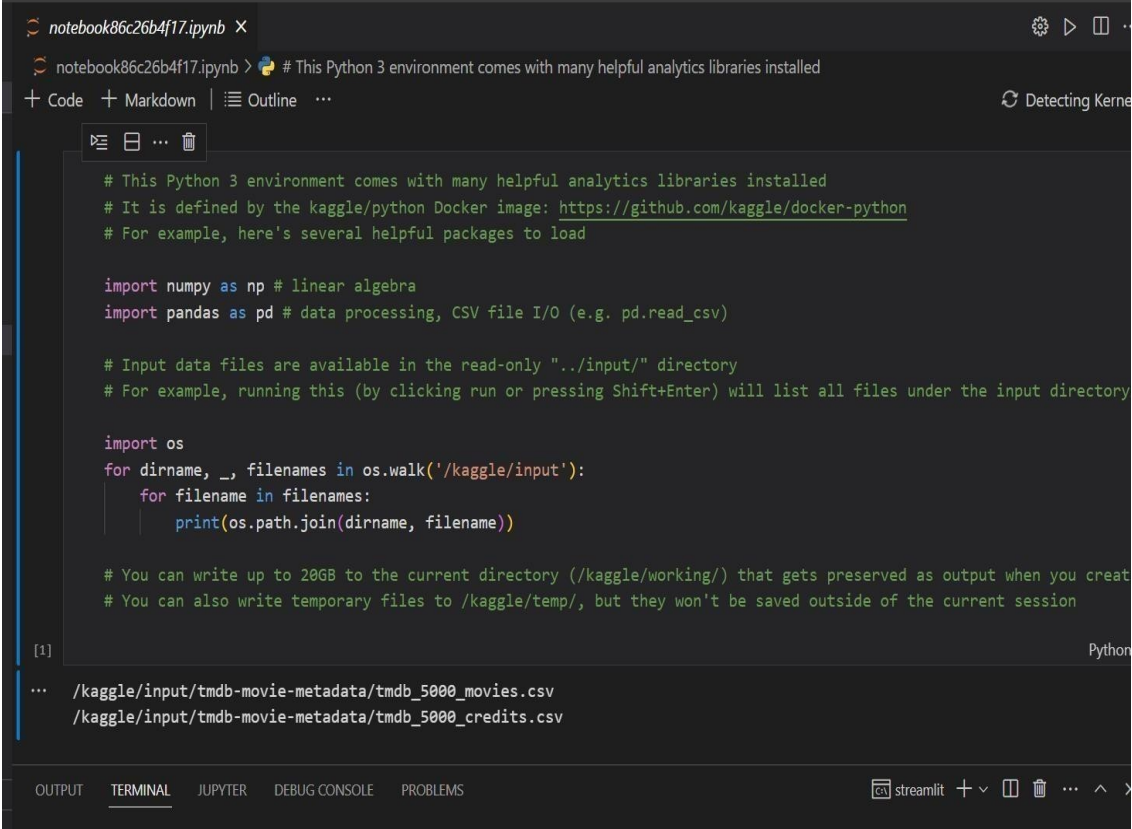
$$A = \sum_{i=1}^r \sigma_i u_i v_i^T.$$

We first prove a simple lemma stating that two matrices A and B are identical if  $Av = Bv$  for all v. The lemma states that in the abstract, a matrix A can be viewed as a transformation that maps vector v onto Av

## Experimental requirements

### Code: Jupyter notebook

In this project we have used popular python libraries to build an interactive user interface



The screenshot shows a Jupyter Notebook interface with a dark theme. The top bar indicates the notebook is named 'notebook86c26b4f17.ipynb'. Below the top bar, there's a toolbar with icons for running, saving, and other actions. The main area contains a code cell with the following text:

```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output when you creat
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session
```

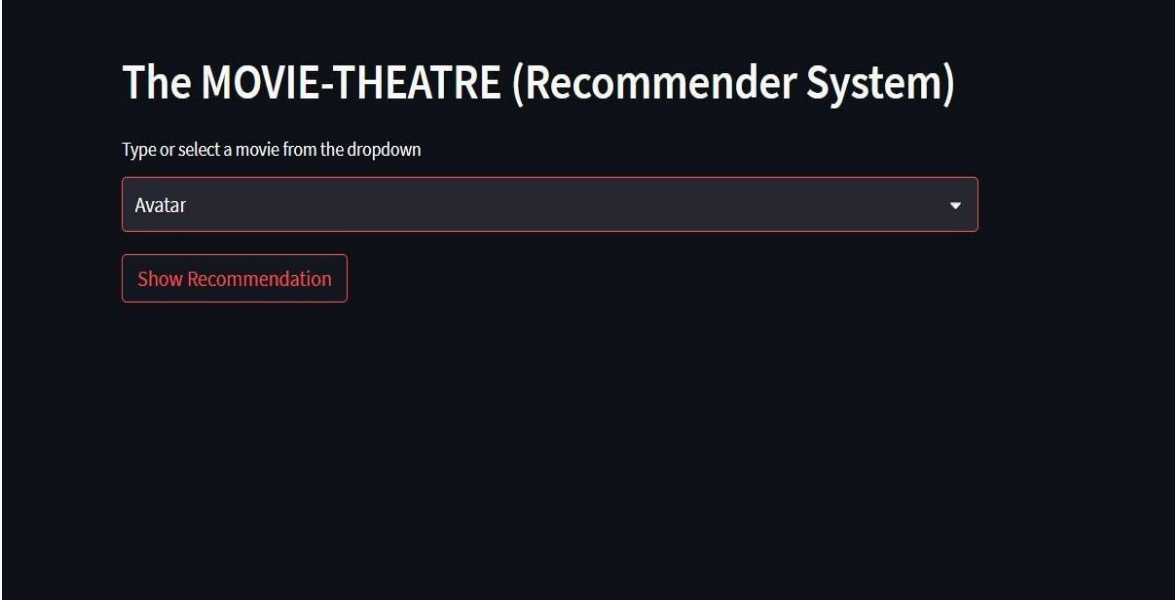
Below the code cell, the output is displayed, showing the paths of the files found in the /kaggle/input directory:

```
... /kaggle/input/tmdb-movie-metadata/tmdb_5000_movies.csv
    /kaggle/input/tmdb-movie-metadata/tmdb_5000_credits.csv
```

The bottom of the interface shows a toolbar with tabs for OUTPUT, TERMINAL, JUPYTER, DEBUG CONSOLE, and PROBLEMS. The TERMINAL tab is currently selected.

Fig:-5.1 Sample Code snippet

For backend we have use database and libraries like pandas , numpy



The screenshot shows a web application titled "The MOVIE-THEATRE (Recommender System)". Below the title, there is a prompt "Type or select a movie from the dropdown". A dropdown menu is displayed with "Avatar" selected. Below the dropdown is a button labeled "Show Recommendation". The entire interface is set against a dark blue background.

Fig:-5.2 Backed code snippet

We have developed our machine learning model in python .

By using flask, we generate resulting api which stores the data in the form of json format these data is retrieved in react by using axios npm mode and then displaying the data

## RESULTS 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 combiningboth.

In our project we have developed a hybrid approach i.e combination of both content and collaborative filtering .Both the approaches have advantages and dis-advantages .in content based filtering the it 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

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

Fig:-6.1 Comparison between the three approaches

The main disadvantage in hybrid approach is it require high memory

**Screen shot** of the result:

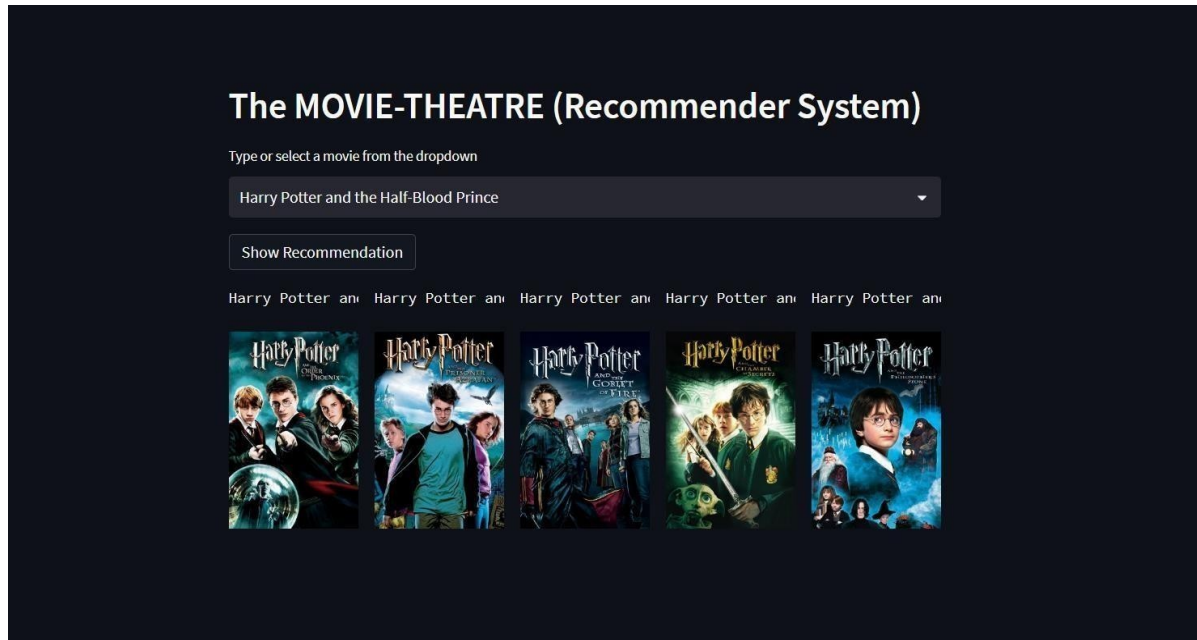


Fig:-6.2 Display of list of recommended movies

Once the name is entered the list of recommended movies are displayed



# TESTING

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that all the system elements have been properly integrated and perform allocated functions. The testing process is actually carried out to make sure that the product exactly does the same thing what is supposed to do. In the testing stage following goals are tried to achieve: -

To affirm the quality of the project.

To find and eliminate any residual errors from previous stages.

To validate the software as a solution to the original problem.

To provide operational reliability of the system.

## 7.1 Testing Methodologies

There are many different types of testing methods or techniques used as part of the software testing methodology. Some of the important testing methodologies are:

### 7.1.1 Unit Testing

Unit testing is the first level of testing and is often performed by the developers themselves. It is the process of ensuring individual components of a piece of software at the code level are functional and work as they were designed to. Developers in a test-driven environment will typically write and run the tests prior to the software or feature being passed over to the test team. Unit testing can be conducted manually, but automating the process will speed up delivery cycles and expand test coverage.

Unit testing will also make debugging easier because finding issues earlier means they take less time to fix than if they were discovered later in the testing process. Test Left is a tool that allows advanced testers and developers to shift left with the fastest test automation tool embedded in any IDE.

## **Integration Testing**

After each unit is thoroughly tested, it is integrated with other units to create modules or components that are designed to perform specific tasks or activities. These are then tested as group through integration testing to ensure whole segments of an application behave as expected (i.e, the interactions between units are seamless). These tests are often framed by user scenarios, such as logging into an application or opening files.

Integrated tests can be conducted by either developers or independent testers and are usually comprised of a combination of automated functional and manual tests.

## **System Testing**

System testing is a black box testing method used to evaluate the completed and integrated system, as a whole, to ensure it meets specified requirements. The functionality of the software is tested from end-to-end and is typically conducted by a separate testing team than the development team before the product is pushed into production

# CONCLUSION AND FUTURE SCOPE

## 8.1 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.

## 8.2 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

- Hirdesh Shivhare, Anshul Gupta and Shalki Sharma (2015), “Recommender system using fuzzy c-means clustering and genetic algorithm based weighted similarity measure”, IEEE International Conference on Computer, Communication and Control.
- Manoj Kumar, D.K. Yadav, Ankur Singh and Vijay Kr. Gupta (2015), “A Movie Recommender System: MOVREC”, International Journal of Computer Applications (0975 – 8887) Volume 124 – No.3.
- RyuRi Kim, Ye Jeong Kwak, HyeonJeong Mo, Mucheol Kim, Seungmin Rho, Ka Lok Man, Woon Kian Chong (2015), “Trustworthy Movie Recommender System with Correct Assessment and Emotion Evaluation”, Proceedings of the International MultiConference of Engineers and Computer Scientists Vol II.
- Zan Wang, Xue Yu\*, Nan Feng, Zhenhua Wang (2014), “An Improved Collaborative Movie Recommendation System using Computational Intelligence”, Journal of Visual Languages & Computing, Volume 25, Issue 6.
- Debadrita Roy, Arnab Kundu, (2013), “Design of Movie Recommendation System by Means of Collaborative Filtering”, International Journal of Emerging Technology advanced Engineering Volume , Issue

