MOVIE RECOMMENDATION SYSTEM

Project Submitted in Partial Fulfillment of the Requirements for the Degree of Bachelor of Technology in the field of Computer Science and Engineering

BY
ABHISHEK SINGH (123200903003)
SAKSHEE (123200903106)
AGNIBHA MAITY(123200903011)

Under the supervision
of
MR. SANKET DAN

Department of Computer Science and Engineering
JIS College of Engineering

Block-A, Phase-III, Kalyani, Nadia, Pin-741235
West Bengal, India

May, 2024

JIS College of Engineering
Block 'A', Phase-III, Kalyani, Nadia, 741235
Phone: +91 33 2582 2137, Telefax: +91 33 2582 2138
Website: www.jiscollege.ac.in, Email: info@jiscollege.ac.in

CERTIFICATE

This is to certify that **SAKSHEE(123200903106), ABHISHEK SINGH (123200903003), AGIBHA MAITY (123200903011).** has completed their project entitled **Movie Recommendation System,** under the guidance of **Mr. Sanket Dan** in partial fulfilment of the requirements for the award of the **Bachelor of Technology in Computer Science and Engineering** from JIS college of Engineering (An Autonomous Institute)is an authentic record of their own work carried out during the academic year 2023-2024 and to the best of our knowledge, this work has not been submitted elsewhere as part of the process of obtaining a degree, diploma, fellowship or any other similar title.

--------------------------------	------------------------------	----------------------------
Signature of the Supervisor	**Signature of the HOD**	**Signature of the Principal**

Place: KALYANI

Date: 21/04/2024

ACKNOWLEDGEMENT

The analysis of the project work wishes to express our gratitude to Mr. Sanket Dan for allowing the degree attitude and providing effective guidance in development of this project work. His conscription of the topic and all the helpful hints, he provided, contributed greatly to successful development of this work, without being pedagogic and overbearing influence. We also express our sincere gratitude to Dr. Bikramjit Sarkar, Head of the Department of Computer Science and Engineering of JIS College of Engineering and all the respected faculty members of Department of CSE for giving the scope of successfully carrying out the project work. Finally, we take this opportunity to thank to Prof. **(Dr.) Partha Sarkar**, Principal of JIS College of Engineering for giving us the scope of carrying out the project work.

Date:

Team members:

ABHISHEK SINGH
B.TECH in Computer Science And Engineering
4^{th} Year /8^{th} SEMESTER
Univ Roll-123200903003

SAKSHEE
B.TECH in Computer Science And Engineering
4^{th} Year /8^{th} SEMESTER
Univ Roll-123200903106

AGNIBHA MAITY
B.TECH in Computer Science And Engineering
4^{th} Year /8^{th} SEMESTER
Univ Roll-123200903011

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 both methods.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

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.

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 least time and to reduce computation time of the movie recommender engine we used cosine similarity measure.

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.

CHAPTER 2

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. Content based [1], [2] collaborative [3] and hybrid [4] are the different approaches used by past researcher for the development of recommender system. In 2007 a web-based movie recommendation system using hybrid filtering methods is presented by the authors [5]. In 2011 a movie recommendation system based on genre correlations is proposed by the authors [6].In 2013 a Bayesian network and Trust model based movie recommendation system is proposed, the Bayesian network is imported for user preference modeling and trust model is used to filter the recommending history data and enable the system to tolerant the noisy data [7].In 2016, authors proposed Recommender systems to predict the rating for users and items, predominantly from big data to recommend their likes. Movie recommendation systems provide a mechanism to assist users in classifying users with similar interests. This system (K-mean Cuckoo) has 0.68 MAE [8], [9].In 2017 authors used a new approach that can solve sparsity problem to a great extent[10].In 2018, authors built a recommendation engine by analyzing rating data sets collected from Twitter to recommend movies to specific user using R[39].

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.

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.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.

CHAPTER 3

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.3

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.

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

CHAPTER 4

SYSTEM ANALYSIS AND DESIGN

4.1 System Architecture of Proposed System:

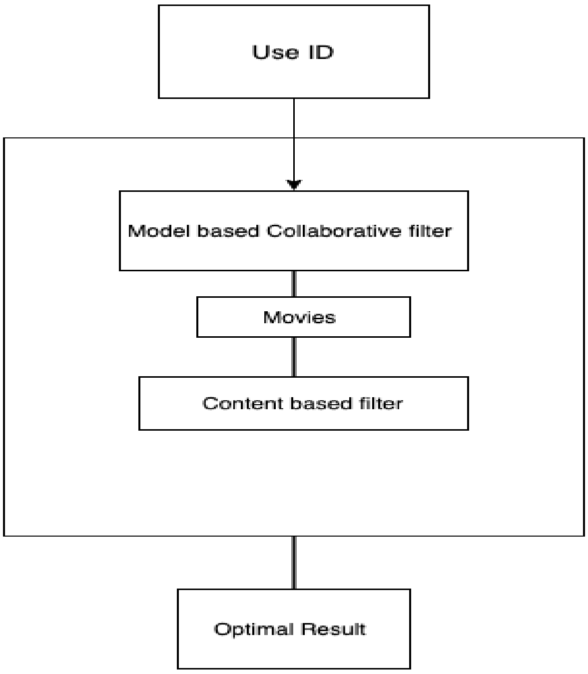

Fig:-4.1 Architecture for hybrid approach

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.

Activity Diagram:

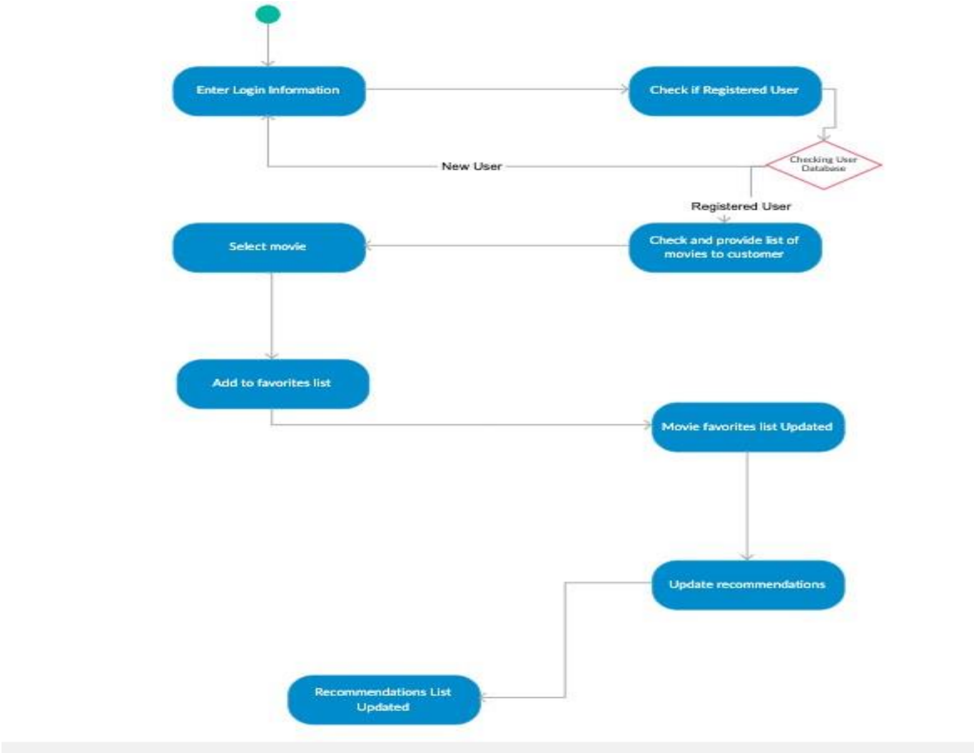

Fig:-4.2 Activity diagram

Once the user login by entering the user-id i.e present in the csv file ranges from 15000 the list of movie are recommended to the user .

4.3 Dataflow:

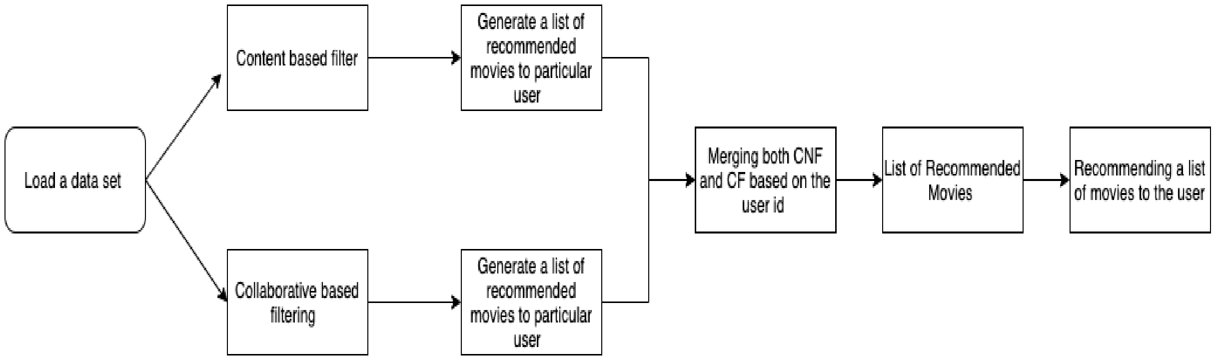

Fig:-4.3 Data Flow Diagram

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

CHAPTER 5

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

5.2 Singular Value Decomposition (SVD):

Let A be an n*d matrix with singular vectors v1, v2, . . . , vr and corresponding singular values σ1, σ2, . . . , σr. Then ui = (1/σi)Avi , for i = 1, 2, . . . , r, are the left singular vectors and by Theorem 1.5, A can be decomposed into a sum of rank one matrices a

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

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

CHAPTER 6

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 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 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

CODE-

File Edit View Insert Cell Kernel Widgets Help | Not Trusted | Python 3 (ipykernel)

Run | Code

In [1]:
```python
import numpy as np
import pandas as pd
```

In [2]:
```python
movies = pd.read_csv('tmdb_5000_movies.csv')
credits = pd.read_csv('tmdb_5000_credits.csv')
```

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

Out[3]:

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

File Edit View Insert Cell Kernel Widgets Help | Not Trusted | Python 3 (ipykernel)

Run | Code

In [9]:
```python
movies.isnull().sum()
```

Out[9]:
```
movie_id    0
title       0
overview    3
genres      0
keywords    0
cast        0
crew        0
dtype: int64
```

In [10]:
```python
movies.dropna(inplace=True)
```

In [11]:
```python
movies.duplicated().sum()
```

Out[11]: 0

In [12]:
```python
import ast
```

In [13]:
```python
def convert(text):
    L = []
    for i in ast.literal_eval(text):
        L.append(i['name'])
    return L
```

In [14]:
```python
movies['genres'] = movies['genres'].apply(convert)
```

In [15]:
```python
movies.head()
```

Out[16]:

	movie_id	title	overview	genres	keywords	cast	crew
0	19995	Avatar	In the 22nd century, a paraplegic Marine is di...	[Action, Adventure, Fantasy, Science Fiction]	[culture clash, future, space war, space colon...	[{"cast_id": 242, "character": "Jake Sully", "...	[{"credit_id": "52fe48009251416c750aca23", "de...
1	285	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	[Adventure, Fantasy, Action]	[ocean, drug abuse, exotic island, east india ...	[{"cast_id": 4, "character": "Captain Jack Spa...	[{"credit_id": "52fe4232c3a36847f800b579", "de...
2	206647	Spectre	A cryptic message from Bond's past sends him o...	[Action, Adventure, Crime]	[spy, based on novel, secret agent, sequel, mi...	[{"cast_id": 1, "character": "James Bond", "cr...	[{"credit_id": "54805967c3a36829b5002c41", "de...
3	49026	The Dark Knight Rises	Following the death of District Attorney Harve...	[Action, Crime, Drama, Thriller]	[dc comics, crime fighter, terrorist, secret i...	[{"cast_id": 2, "character": "Bruce Wayne / Ba...	[{"credit_id": "52fe4781c3a36847f81398c3", "de...
4	49529	John Carter	John Carter is a war-weary, former military ca...	[Action, Adventure, Science Fiction]	[based on novel, mars, medallion, space travel...	[{"cast_id": 5, "character": "John Carter", "c...	[{"credit_id": "52fe479ac3a36847f813eaa3", "de...

In [17]:
```python
def convert3(text):
    L = []
    counter = 0
    for i in ast.literal_eval(text):
        if counter < 3:
            L.append(i['name'])
        counter+=1
    return L
```

In [18]:
```python
movies['cast'] = movies['cast'].apply(convert3)
movies.head()
```

Out[69]: 'in the 22nd century, a parapleg marin is dispatch to the moon pandora on a uniqu mission, but becom torn between follow order and protect an alien civilization. action adventur fantasi sciencefict cultureclash futur spacewar spacecoloni societi spacetravel futurist romanc space alien tribe alienplanet cgi marin soldier battl loveaffair antiwar powerrel mindandsoul 3d samworthington zoesaldana sigourneyweav jamescameron'

In [70]:
```
from sklearn.metrics.pairwise import cosine_similarity
```

In [77]:
```
similarity = cosine_similarity(vectors)
```

In [78]:
```
similarity
```

Out[78]:
```
array([[1.        , 0.08458258, 0.08718573, ..., 0.04559608, 0.        ,
        0.        ],
       [0.08458258, 1.        , 0.06063391, ..., 0.02378257, 0.        ,
        0.02615329],
       [0.08718573, 0.06063391, 1.        , ..., 0.02451452, 0.        ,
        0.        ],
       ...,
       [0.04559608, 0.02378257, 0.02451452, ..., 1.        , 0.03962144,
        0.04229549],
       [0.        , 0.        , 0.        , ..., 0.03962144, 1.        ,
        0.08714204],
       [0.        , 0.02615329, 0.        , ..., 0.04229549, 0.08714204,
        1.        ]])
```

In [92]:
```
def recommend(movie):
    movie_index = new_df[new_df['title'] == movies].index[0]
    distances = similarity[movie_index]
    movies_list = sorted(list(enumerate(distances)),reverse=True,key = lambda x: x[1])[1:6]

    for i in movies_list:
        print(new_df.iloc[i[0]].title)
```


```
        0.04229549],
       [0.        , 0.        , 0.        , ..., 0.03962144, 1.        ,
        0.08714204],
       [0.        , 0.02615329, 0.        , ..., 0.04229549, 0.08714204,
        1.        ]])
```

In [92]:
```
def recommend(movie):
    movie_index = new_df[new_df['title'] == movies].index[0]
    distances = similarity[movie_index]
    movies_list = sorted(list(enumerate(distances)),reverse=True,key = lambda x: x[1])[1:6]

    for i in movies_list:
        print(new_df.iloc[i[0]].title)
```


In [98]:
```
recommend('Batman Begins')
```

```
C:\Users\LENOVO\AppData\Local\Temp\ipykernel_22144\701328947.py:2: FutureWarning: Automatic reindexing on DataFrame vs Series comparisons is deprecated and will raise ValueError in a future version. Do `left, right = left.align(right, axis=1, copy=False)` before e.g. `left == right`
  movie_index = new_df[new_df['title'] == movies].index[0]
```

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

In [99]:
```
import pickle
```

In [97]:
```
pickle.dump(new_df,open('movie_list.pkl','wb'))
```

TABLE 1:

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

A. Comparison with Existing Technology : The table 2 and 3 compares the result of the proposed system with the existing technique. These tables shows a comparison of RMSE with the existing technique i.e. cuckoo search. It is seen from the tables that for the existing technique the RMSE value is 1.23154 for cluster equal to 68, RMSE value using proposed technique is 1.233 to 19 clusters and RMSE value using proposed technique is 1.081648 to 2 clusters

TABLE II: RMSE in Proposed Technique

Root Mean Squared Error	No. of Cluster
1.23154	68

TABLE III: RMSE in Existing Technique

Root Mean Squared Error	No. of Cluster
1.2333	19
1.081648	2

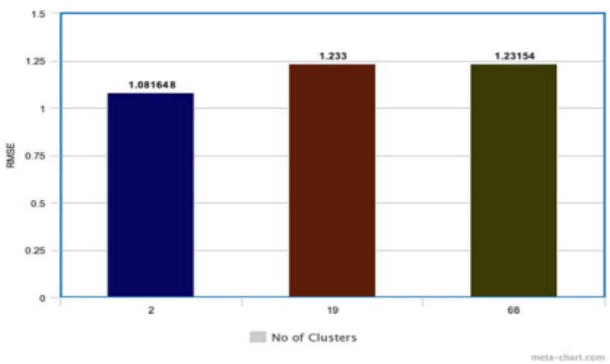

Fig. 9: Comparison Graph with the Existing Technique

CHAPTER 7

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:

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.

CHAPTER 8

CONCLUSION AND FUTRURE 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 –

[1] Goel A., Khandelwal D., Mundhra J., Tiwari R. (2018) Intelligent and Integrated Book Recommendation and Best Price Identifier System Using Machine Learning. In: Bhateja V., Coello Coello C., Satapathy S., Pattnaik P. (eds) Intelligent Engineering Informatics. Advances in Intelligent Systems and Computing, vol 695. Springer, Singapore .

[2] Bao J., Zheng Y. (2017) Location-Based Recommendation Systems. In: Shekhar S., Xiong H., Zhou X. (eds) Encyclopedia of GIS. Springer, Cham .

[3] Chavarriaga O., Florian-Gaviria B., Solarte O. (2014) A Recommender System for Students Based on Social Knowledge and Assessment Data of Competences. In: Rensing C., de Freitas S., Ley T., Munoz-Merino ˜ P.J. (eds) Open Learning and Teaching in Educational Communities. EC-TEL 2014. Lecture Notes in Computer Science, vol 8719. Springer, Cham .

[4] F.O.Isinkaye et. al, Recommendation systems: Principles, methods and evaluation, Egyptian Informatics Journal Volume 16, Issue 3, November 2015.

[5] H. Drachsler, T. Bogers, R. Vuorikari, K. Verbert, E. Duval, N. Manouselis, G. Beham, S. Lindstaedt, H. Stern, M. Friedrich, et al. Issues and considerations regarding sharable data sets for recommender systems in technology enhanced learning. Procedia Computer Science, 1(2):2849–2858, 2010.

[6] AlvaroTejeda-Lorente, A quality based recommender system to dissem- ´ inate information in a university digital library,Information Sciences Volume 261, 10 March 2014, Pages 52-69 .

[7] Trang Tran, T.N., Atas, M., Felfernig, A. et al. J Intell Inf Syst (2018) 50: 501. https://doi.org/10.1007/s10844-017-0469-0 .

[8] Xin Luo, Mengchu Zhou, Yunni Xia, and Qingsheng Zhu,An Efficient Non-Negative Matrix-Factorization-Based Approach to Collaborative Filtering for Recommender Systems,IEEE Transactions on Industrial Informatics (Volume: 10 , Issue: 2 , May 2014) .

[9] Badsha, S., Yi, X. Khalil, I. Data Sci. Eng. (2016) 1: 161. https://doi.org/10.1007/s41019-016-0020-2 .

[10] Farman Ullah, Ghulam Sarwar, Sung Chang Lee, Yun Kyung Park, Kyeong Deok Moon, Jin Tae Kim, Hybrid Recommender System with Temporal Information,The International Conference on Information Network, 2012,DOI: 10.1109/ICOIN.2012.6164413.