VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI – 590 018

An Internship Project Report
on

Movie Recommendation System

Submitted in partial fulfilment of the requirements for the VII Semester of the degree of **Bachelor of Engineering in Information Science and Engineering** of Visvesvaraya Technological University, Belagavi

by

Pavithra Sonu J
1RN18IS073

Under the Guidance of

Dr. S Sathish Kumar
Professor
Department of
ISE

ESTD:2001
An Institute with a Difference

Department of Information Science and Engineering

RNS Institute of Technology

Dr. Vishnuvaradhan Road, Rajarajeshwari Nagar post,
Channasandra, Bengaluru-560098

2021-2022

RNS INSTITUTE OF TECHNOLOGY

Dr. Vishnuvaradhan Road, Rajarajeshwari Nagar post,
Channasandra, Bengaluru - 560098

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the Internship work entitled ***Movie Recommendation System*** has been successfully completed by **Pavithra Sonu J (1RN18IS073)** a bonafide student of **RNS Institute of Technology, Bengaluru** in partial fulfilment of the requirements of 7th semester for the award of degree in **Bachelor of Engineering in Information Science and Engineering** of **Visvesvaraya Technological University, Belagavi** during the academic year **2021-2022**. The internship report has been approved as it satisfies the academic requirements in respect of internship work for the said degree.

Dr. S Sathish Kumar	**Dr. Suresh L**	**Dr. M K Venkatesha**
Internship Guide	Professor and HoD	Principal
Professor	Department of ISE	RNSIT
Department of ISE	RNSIT	

External Viva

Name of the Examiners	**Signature with Date**
1. ____________________________	**1.**____________________ ___
2. ____________________________	**2.**____________________ ___

DECLARATION

I, **PAVITHRA SONU J [USN: 1RN18IS073]** student of VII Semester BE, in Information Science and Engineering, RNS Institute of Technology hereby declare that the Internship work entitled ***Movie Recommendation System*** has been carried out by me and submitted in partial fulfillment of the requirements for the VII Semester degree of ***Bachelor of Engineering in Information Science and Engineering*** of Visvesvaraya Technological University, Belagavi during the academic year 2021-2022.

Place: Bengaluru

Date: 9th January 2022

PAVITHRA SONU J

(1RN18IS073)

ABSTRACT

Nowadays, the recommendation system has made finding the things easy that we need. Movie recommendation system aim at helping the user by suggesting what movie to watch without having to go through the long process of choosing from a large set of movies which go up to thousands and millions that is time consuming and confusing. In this project, the aim is to reduce the human effort by suggesting movies based on the user's interests. To handle such problems, we introduced a model combining both content-based and collaborative approach which will give progressively explicit outcomes. It will give progressively explicit outcomes compared to different systems that are based on content-based approach. Content-based recommendation systems are constrained to people, these systems don't prescribe things out of the box, thus limiting your choice to explore more. Hence, it is focused on a system that resolves these issues. It analyzes the correlation with similar movies.

ACKNOWLEDGMENT

At the very onset, I would like to place our gratefulness to all those people who helped me in making the Internship a successful one.

Coming up, this internship to be a success was not easy. Apart from the sheer effort, the enlightenment of the very experienced teachers also plays a paramount role because it is they who guided me in the right direction.

First, I would like to thank the **Management of RNS Institute of Technology** for providing such a healthy environment for the successful completion of internship work.

In this regard, I express sincere gratitude to our beloved Principal **Dr. M K Venkatesha,** for providing us with all the facilities.

I extremely grateful to my own and beloved Professor and Head of the Department of Information science and Engineering, **Dr. Suresh L**, for having accepted to patronize me in the right direction with all his wisdom.

I place my heartfelt thanks to **Dr. S Sathish Kumar** Professor, Department of Information Science and Engineering for having guided the internship and all the staff members of the Department of Information Science and Engineering for helping at all times.

I thank **Mr. Aman Upadhyay, NASTECH**, for providing the opportunity to be a part of the Internship program and has guided me to complete the same successfully.

I also thank my internship coordinator **Dr. R Rajkumar,** Associate Professor, Department of Information Science and Engineering. I would thank my friends for having supported me with all their strength and might. Last but not the least, I thank my parents for supporting and encouraging me throughout. I have made an honest effort in this assignment.

PAVITHRA SONU J

TABLE OF CONTENTS

Abstract	**iv**
Acknowledgment	**v**
Contents	**vi**
List of figures	**viii**
List of Tables	**ix**
List of Abbreviations	**x**
1. Introduction	**1**
1.1 Background	1
1.2 Existing System	1
1.3 Proposed System	2
2. Literature Review	**3**
3. Analysis	**6**
3.1 Introduction	6
3.2 Software requirement specification	6
3.2.1 Hardware Requirements	7
3.2.2 Software Specifications	7
3.3.3 Software Requirements	7
4. System Design	**9**
4.1 Introduction	9
4.2 System Architecture	9
4.3 High-level Design	10
4.3.1 Dataflow	10
4.3 Low-level Design	10
4.3.1 Activity Diagram	10

5. Implementation Details **11**

6. Testing **14**

7. Results **16**

8. Conclusion and Future Enhancement **19**

8.1 Conclusion 19

8.2 Future Enhancement 19

9. References **20**

LIST OF FIGURES

Figure. No.	**Descriptions**	**Page**
Figure. 4.2	Architecture of hybrid approach	09
Figure. 4.3.1	Dataflow	10
Figure. 4.3.2	Activity Diagram	10
Figure. 5.1	Importing libraries and dataset	11
Figure. 5.1.1	Checking all data	11
Figure 5.2	Mean rating and count rating of all movies	11
Figure. 5.3	Create Dataframe	11
Figure. 5.4	Plot graph	12
Figure. 5.5	Sorting values	12
Figure. 5.6	Analyzing correlation	13
Figure. 5.7	Similar movies	13
Figure. 7.4.1	Graph of no. of rating of columns	18
Figure. 7.4.2	Graphical representation of rating columns	18

LIST OF TABLES

Table. No.	Descriptions	Page
Table. 7.1	Head of data	16
Table. 7.1	Check movies with IDs	16
Table. 7.3	Count rating table	17
Table. 7.4	Dataframe table	17
Table. 7.6	Similar movies	18

ABBREVIATIONS

ML	-	Machine Learning
AI	-	Artificial Intelligence
GPU	-	Graphics Processing Units
API	-	Application Programming Interface
KNN	-	K Nearest Neighbors

Chapter 1

INTRODUCTION

1.1 Background

Collaborative filtering:

Collaborative filtering approaches build a model from the user’s past behavior (i.e. items purchased or searched by the user) as well as similar decisions made by other users. This model is then used to predict items (or ratings for items) that users may have an interest in.

There are two kinds of collaborative filtering systems; user-based recommender and item-based recommender.

Content-based filtering:

Content-based filtering approaches uses a series of discrete characteristics of an item in order to recommend additional items with similar properties. Content-based filtering methods are totally based on a description of the item and a profile of the user’s preferences. It recommends items based on the user’s past preferences. This project is to provide accurate movie recommendations to users.

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.

1.2 Existing System

Traditional movie websites (IMDB) function by proving global user ratings on movies in their database. Movies are categorized based on genre, era, directors and so on. Users can search for movies, browse lists and read reviews written by critics or other users. However, most of these services lack any personal recommendation system and haven’t taken advantage of social networking communities. Also draws upon its vast user base to give lists of similar movie fans, their ratings and reviews. Providing a basic recommendation system by suggesting items that are most similar to a particular item, in this case, movies. It just tells what movies/items are most similar to the user’s movie choice.

1.3 Proposed System

This section contains a series of steps and the methodology of the proposed system. How the system is going to operate, and events that are going to occur. 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.

Chapter 2

LITERATURE REVIEW

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.

Literature review of recommendation systems:

Kumar et al. [29] proposed MOVREC, a movie recommendation system based on collaborative filtering approaches. Collaborative filtering takes the data from all the users and based on that generates recommendations. A hybrid system has been presented by Virk et al. [30]. This system combines both collaborative and content-based method. De Campos et al. [34] also made an analysis of both the traditional recommendation techniques. As both of these techniques have certain setbacks, he proposed another system which is a combination of Bayesian network and collaborative technique. Kużelewska [35] proposed clustering as an approach to handle the recommendations. Two methods for clustering were analyzed: Centroid-based solution and memory-based methods. The result was that accurate recommendations were generated. Chiru et al. [27] proposed Movie Recommender, a system that uses the user's history in order to generate recommendations.

Sharma and Maan [36] in their paper analyzed various techniques used for recommendations, collaborative, hybrid and content-based recommendations. Also, it describes the pros and cons of these approaches.

Li and Yamada [37] proposed an inductive learning algorithm. Here a tree had been built which shows the user recommendation. Some of the major contribution in recommendation system.

Scharf & Alley [38] 1993 The authors proposed a flexible multicomponent rate recommendation system to predict the optimum rate of fertilizer for winter wheat. Basu et al. [39] 1998 The authors proposed an approach to the recommendation that can exploit both ratings and content information. Sarwar et al. [40] 2001 The authors proposed various techniques for computing item similarities. Bomhardt [41] 2004 The author proposed an approach for a personal recommendation of news. Manikrao & Prabhakar [42] 2005 The authors presented the design of a dynamic web selection framework. Von Reischach et al. [43] 2009 The authors proposed a rating concept that allows users to generate rating criteria. Choi et al. [44] 2012 The authors proposed approaches for integrating various techniques for improving the recommendation quality.

Discussed the contribution of filtering techniques for different purposes.
Literature review of filtering techniques:
Goldberg et al. [45] 1992 The authors introduced the collaborative filtering technique. Herlocker et al. [46] 1997 Authors applied filtering techniques to Usenet news. Miyahara & Pazzani [47] 2000 The authors introduced an approach to calculate the similarity between a user from negative ratings to positive ratings separately. Hofmann [48] 2004 The author introduced a new-family of model-based algorithms. Dabov et al. [49] 2008 The authors proposed an image restoration technique using collaborative filtering. Pennock et al. [50] 2013 The authors proposed various approaches for filtering by personality diagnosis. Liu et al. [51] 2014 The authors introduced a new method to provide an accurate recommendation.
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. 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.

One undertaking from the above discussions is that recommendations systems have gained vital name and recognition among researches because of their frequent look in varied and widespread applications within the fields of various branches of science and technology.
The previous recommendation systems had certain gaps in them:

- As most of the users do not provide ratings, the rating matrix becomes very sparse.
- Over-specialization is the most common problem faced by content-based recommendation.
- Content-based recommendation systems always face the problem of a cold start.
- Therefore, this motivates us to provide a new model for society:
- Improves sparsity by making rating as mandatory.
- The problem of over-specialization is resolved using neighborhood-based collaborative techniques.

Movie Recommendation System Using Collaborative Filtering:

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

Collaborative filtering systems analyze the user's behavior 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 behavior. 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 neighbors. 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, itembased 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

ANALYSIS

3.1 Introduction

A recommendation system 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.

Providing a basic recommendation system by suggesting items that are most similar to a particular item, in this case, movies. It just tells what movies/items are most similar to the user’s movie choice.

These systems collect information about a user's preferences and behavior, and then use this information to improve their suggestions in the future.

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 favorable movies.

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.

3.2 Software Requirement Specification

The requirement that the system needs is categorized into hardware and software requirements. These requirements are listed below

3.2.1 Hardware Requirements

- A PC with Windows/Linux OS
- Processor: Intel i5 or more
- Minimum of 8gb RAM
- 2gb Graphic Card

3.2.2 Software Specifications

- Operating system: Windows 7/8/10/11 or Linux
- Google Colab Network
- Python libraries

3.3.3 Software Requirements

What is Colaboratory?

Colaboratory, or “Colab” for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. More technically, Colab is a hosted Jupyter notebook service that requires no setup to use, while providing free access to computing resources including GPUs.

Colab notebooks are Jupyter notebooks that run in the cloud and are highly integrated with Google Drive, making them easy to set up, access, and share. The following sections describe deploying Earth Engine in Google Colab and visualizing maps and charts using third-party Python packages.

Python libraries: For the computation and analysis we need certain python libraries which are used to perform analytics. Packages such as Pandas, Matplotlib, Seaborn are needed.

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.

Matplotlib: It is a collection of functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.

In **matplotlib.pyplot** various states are preserved across function calls, so that it keeps track of things like the current figure and plotting area, and the plotting functions are directed to the current axes (please note that "axes" here and in most places in the documentation refers to the axes part of a figure and not the strict mathematical term for more than one axis).

Seaborn: Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and colour palettes to make statistical plots more attractive. It is built on the top of matplotlib library and also closely integrated to the data structures from pandas.

Seaborn aims to make visualization the central part of exploring and understanding data. It provides dataset-oriented APIs, so that we can switch between different visual representations for same variables for better understanding of dataset.

Chapter 4

SYSTEM DESIGN

4.1 Introduction

Systems design is the process of defining elements of a system like modules, architecture, components and their interfaces and data for a system based on the specified requirements. It is the process of defining, developing and designing systems which satisfies the specific needs and requirements of a business or organization.

The purpose of the System Design process is to provide sufficient detailed data and information about the system and its system elements to enable the implementation consistent with architectural entities as defined in models and views of the system architecture.

4.2 System Architecture

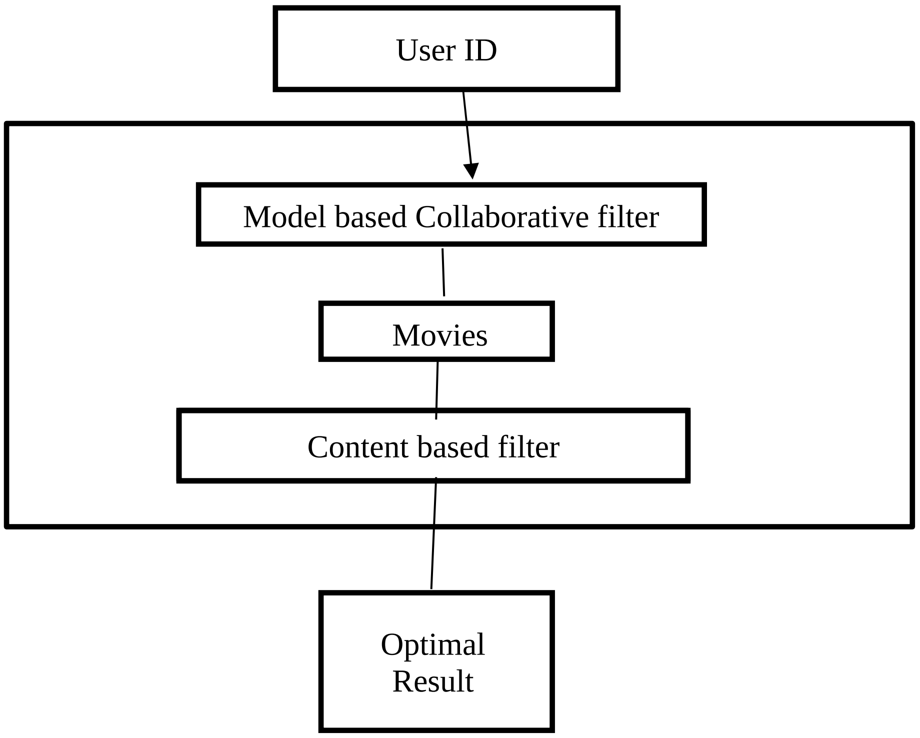


Fig 4.2: Architecture of Hybrid approach

For each different individual use different list of movies are recommended, as 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.

4.3 High Level Design

4.3.1 Data flow

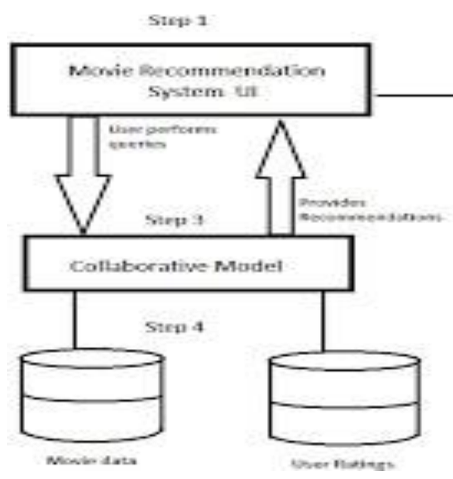


Figure 4.3.1: Data flow

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.

4.4 Low Level Design

4.4.1 Activity Diagram

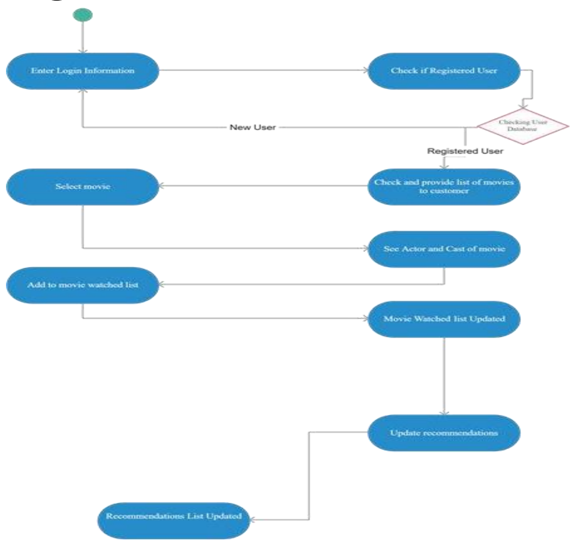

Enter Login Information
Check if Registered User
Checking User Database
New User
Registered User
Select movie
Check and provide list of movies to customer
See Actor and Cast of movie
Add to movie watched list
Movie Watched list Updated
Update recommendations
Recommendations List Updated

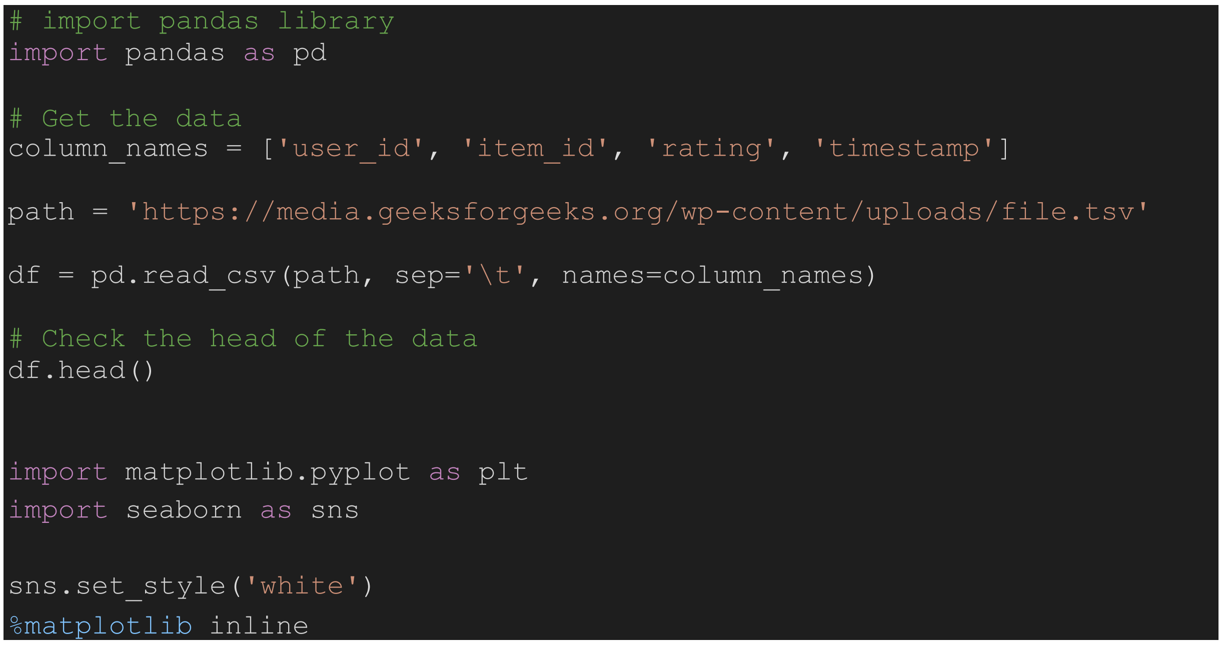
Figure 4.4.1: Activity diagram

Once the user login by entering the user id that is present in the csv file ranges many number of lists of movie are recommended to the user.

Chapter 5

IMPLEMENTATION DETAILS

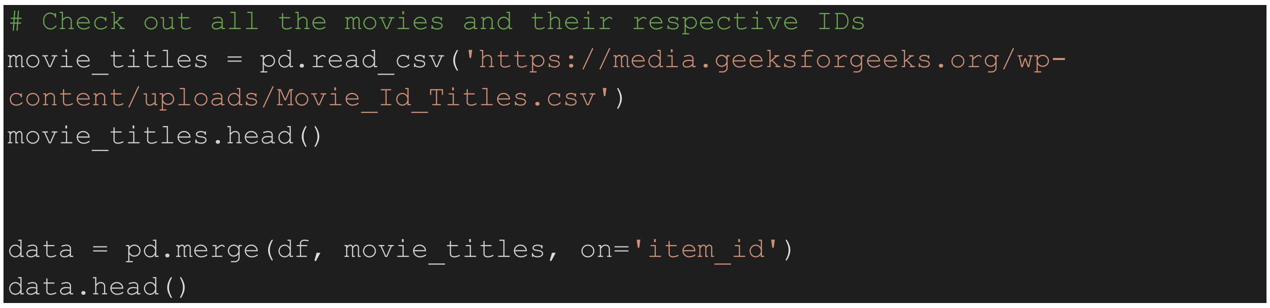
5.1 Importing libraries



```
# import pandas library
import pandas as pd

# Get the data
column_names = ['user_id', 'item_id', 'rating', 'timestamp']

path = 'https://media.geeksforgeeks.org/wp-content/uploads/file.tsv'

df = pd.read_csv(path, sep='\t', names=column_names)

# Check the head of the data
df.head()


import matplotlib.pyplot as plt
import seaborn as sns

sns.set_style('white')
%matplotlib inline
```

Figure 5.1: importing libraries and datasets

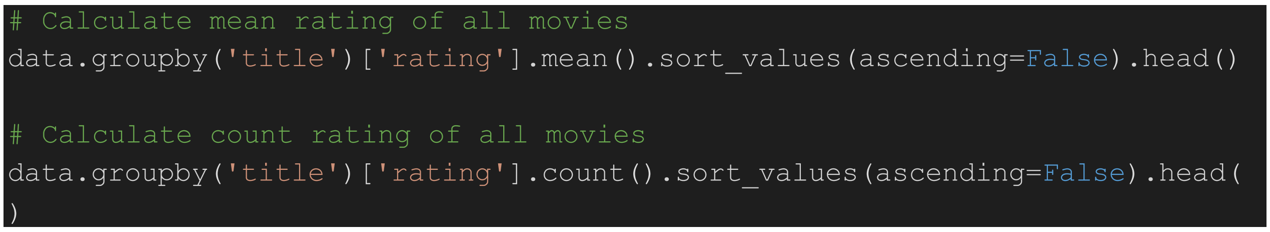
5.1.1 Checking data



```
# Check out all the movies and their respective IDs
movie_titles = pd.read_csv('https://media.geeksforgeeks.org/wp-
content/uploads/Movie_Id_Titles.csv')
movie_titles.head()


data = pd.merge(df, movie_titles, on='item_id')
data.head()
```

Figure 5.1.1: Checking all data

5.2 Mean rating and Count rating of all movies



```
# Calculate mean rating of all movies
data.groupby('title')['rating'].mean().sort_values(ascending=False).head()

# Calculate count rating of all movies
data.groupby('title')['rating'].count().sort_values(ascending=False).head(
)
```

Figure 5.2: Mean rating and Count rating

5.3 Create Dataframe

```
ratings = pd.DataFrame(data.groupby('title')['rating'].mean())

ratings['num of ratings'] =
pd.DataFrame(data.groupby('title')['rating'].count())

ratings.head()
```

Figure 5.3: Create dataframe with 'rating' count values

5.4 Plot Graph

5.4.1 number of ratings column

```
# plot graph of 'num of ratings column'
plt.figure(figsize =(10, 4))

ratings['num of ratings'].hist(bins = 70)
```

Figure 5.4.1: Plot graph for num of ratings column

5.4.2 'rating' column

```
plt.figure(figsize =(10, 4))

ratings['rating'].hist(bins = 70)
```

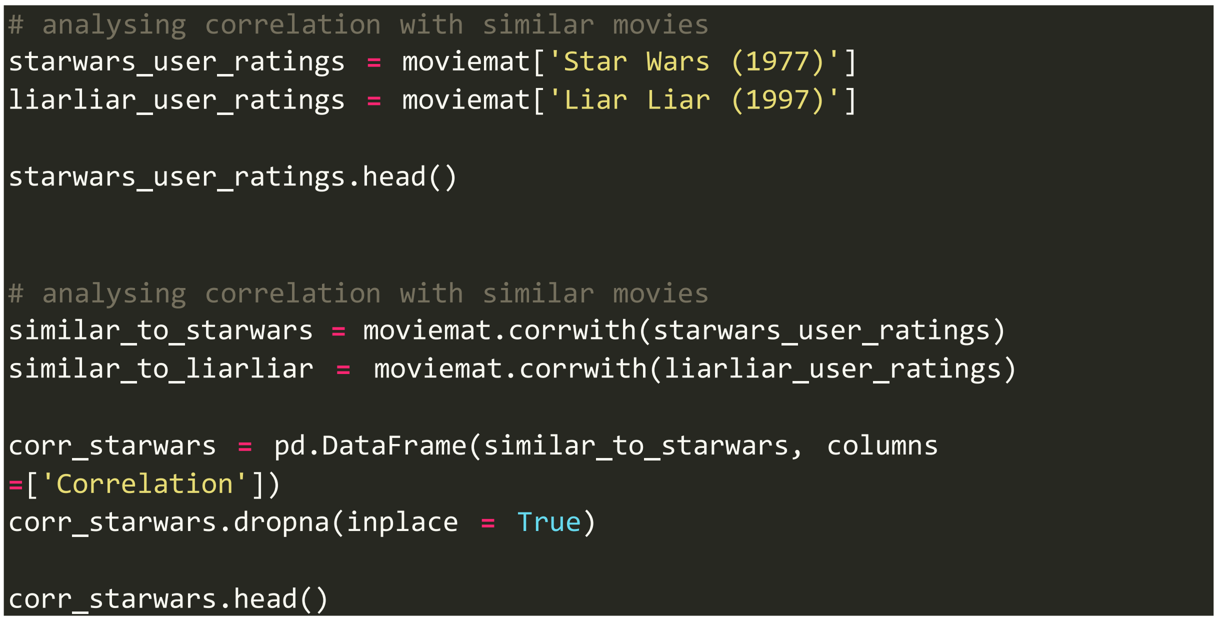
Figure 5.4.2: Plot graph for 'rating' column

5.5 Sorting values

```
# Sorting values according to
# the 'num of rating column'
moviemat = data.pivot_table(index ='user_id',
              columns ='title', values ='rating')

moviemat.head()

ratings.sort_values('num of ratings', ascending = False).head(10)
```

Figure 5.5: Sorting values according to the num of rating column

5.6 Analyzing correlation

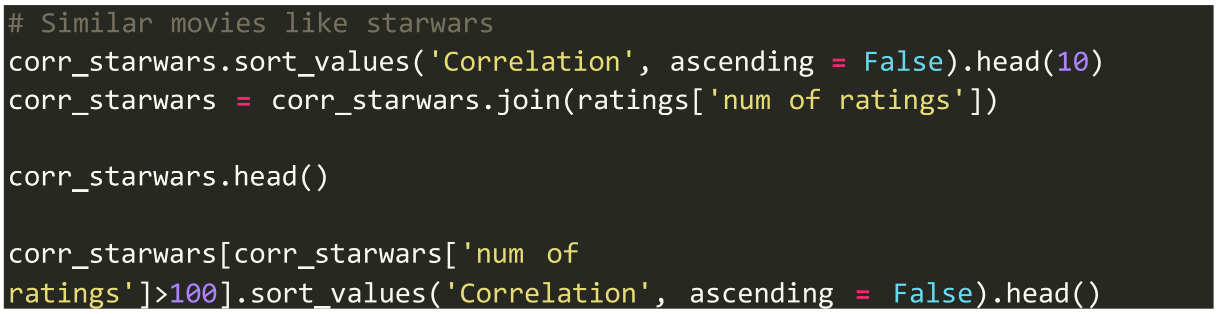


```
# analysing correlation with similar movies
starwars_user_ratings = moviemat['Star Wars (1977)']
liarliar_user_ratings = moviemat['Liar Liar (1997)']

starwars_user_ratings.head()


# analysing correlation with similar movies
similar_to_starwars = moviemat.corrwith(starwars_user_ratings)
similar_to_liarliar  = moviemat.corrwith(liarliar_user_ratings)

corr_starwars = pd.DataFrame(similar_to_starwars, columns
=['Correlation'])
corr_starwars.dropna(inplace = True)

corr_starwars.head()
```

Figure 5.6: Analyzing correlation with similar movies

5.7 Similar movies

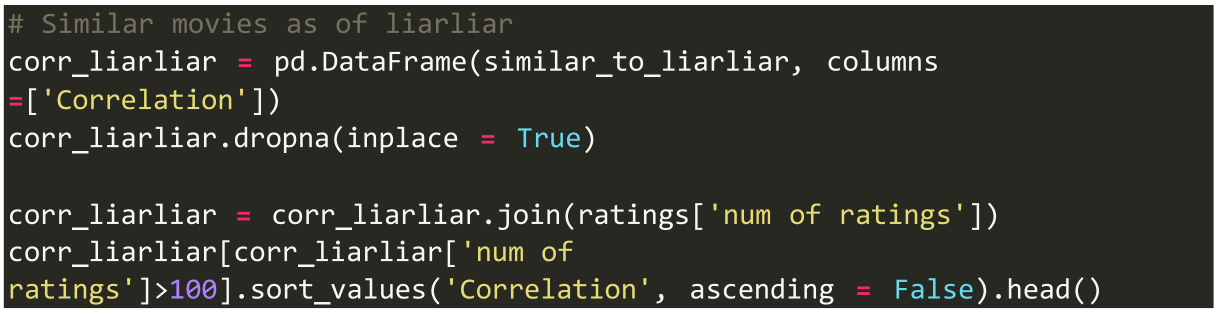
5.7.1 Like starwars



```
# Similar movies like starwars
corr_starwars.sort_values('Correlation', ascending = False).head(10)
corr_starwars = corr_starwars.join(ratings['num of ratings'])

corr_starwars.head()

corr_starwars[corr_starwars['num of
ratings']>100].sort_values('Correlation', ascending = False).head()
```

Figure 5.7.1: Similar movies like starwars

5.7.2 Like liarliar



```
# Similar movies as of liarliar
corr_liarliar = pd.DataFrame(similar_to_liarliar, columns
=['Correlation'])
corr_liarliar.dropna(inplace = True)

corr_liarliar = corr_liarliar.join(ratings['num of ratings'])
corr_liarliar[corr_liarliar['num of
ratings']>100].sort_values('Correlation', ascending = False).head()
```

Figure 5.7.2: Similar movies like liarliar

Chapter 6

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.

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

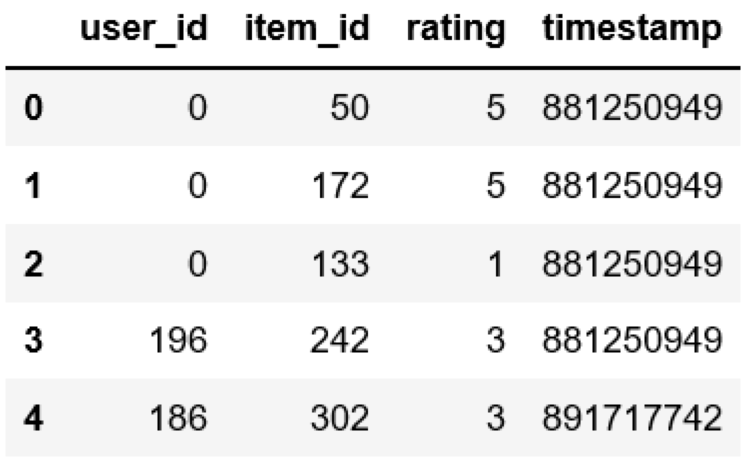
After each unit is thoroughly tested, it is integrated with other units to create modules or components that are designed to perform specific tasks or activities. These are then tested as group through integration testing to ensure whole segments of an application behave as expected (i.e., the interactions 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 type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements. In system testing, integration testing passed componentsare taken as input. 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 7

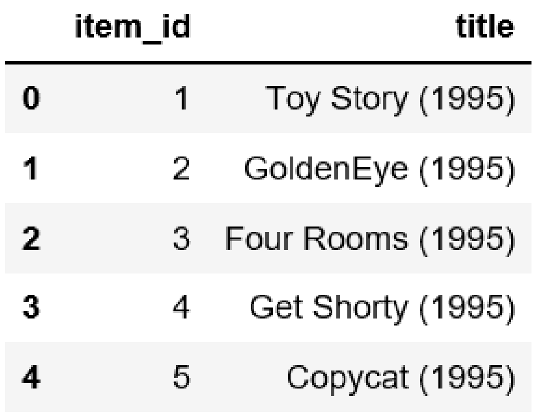
RESULTS



	user_id	item_id	rating	timestamp
0	0	50	5	881250949
1	0	172	5	881250949
2	0	133	1	881250949
3	196	242	3	881250949
4	186	302	3	891717742

Table 7.1: Head of the data

The above table shows the head of the data.

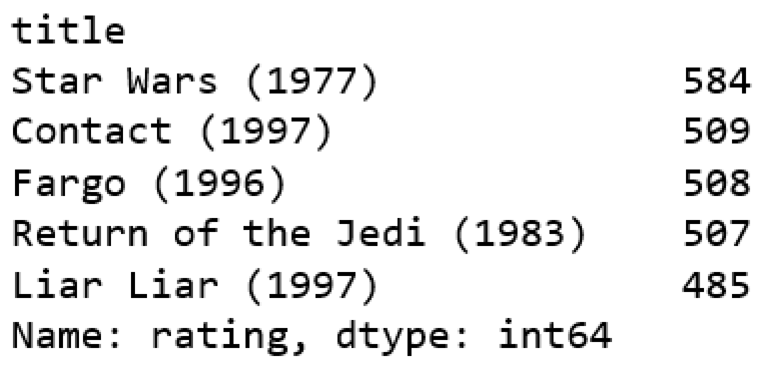


	item_id	title
0	1	Toy Story (1995)
1	2	GoldenEye (1995)
2	3	Four Rooms (1995)
3	4	Get Shorty (1995)
4	5	Copycat (1995)

Table 7.2: check movies with their respective IDs

The above table shows the movies with their IDs.

7.3 Count rating

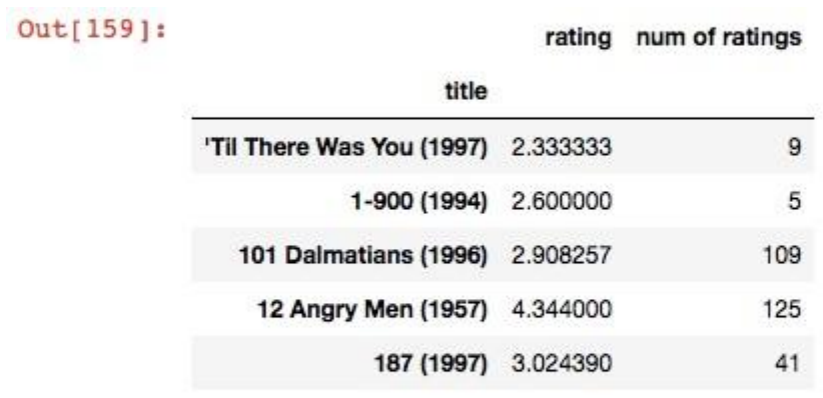


```
title
Star Wars (1977)               584
Contact (1997)                 509
Fargo (1996)                   508
Return of the Jedi (1983)      507
Liar Liar (1997)               485
Name: rating, dtype: int64
```

Table 7.3: count rating of all movies

The above figure shows the count rating of all movies.

7.4 Dataframe



Out[159]:

title	rating	num of ratings
'Til There Was You (1997)	2.333333	9
1-900 (1994)	2.600000	5
101 Dalmatians (1996)	2.908257	109
12 Angry Men (1957)	4.344000	125
187 (1997)	3.024390	41

Table 7.4: Dataframe

The above figure shows the dataframe with rating count values

7.5 Plot graph

```
<matplotlib.axes._subplots.AxesSubplot at 0x1258f8780>
```

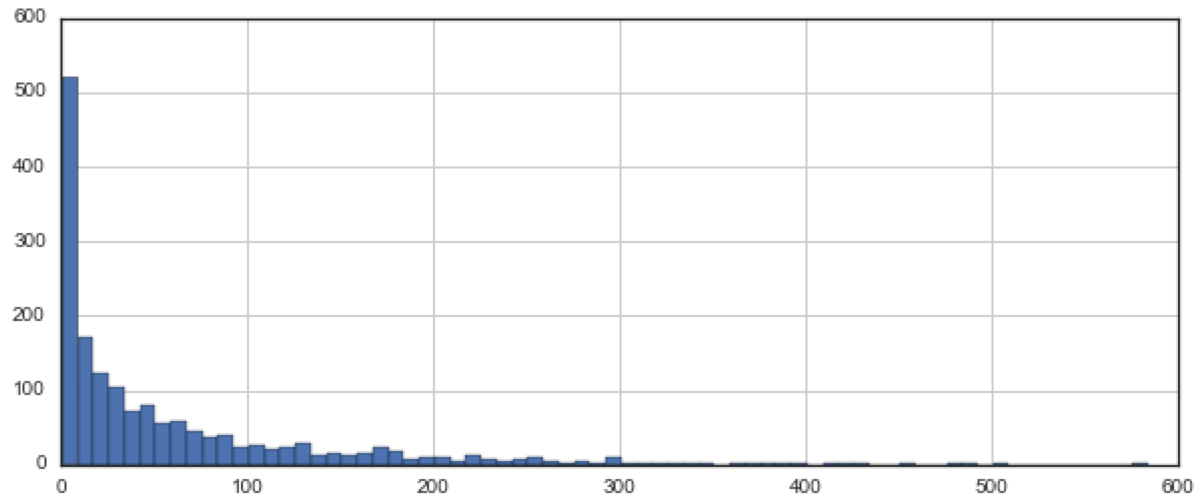


Figure 7.5.1: Graph of num of rating columns

```
<matplotlib.axes._subplots.AxesSubplot at 0x125d12908>
```

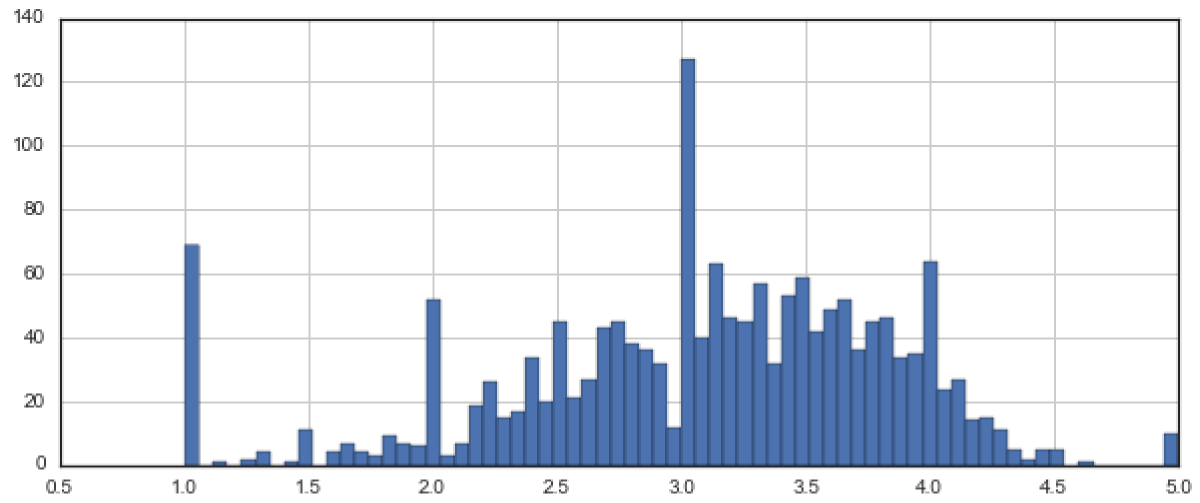
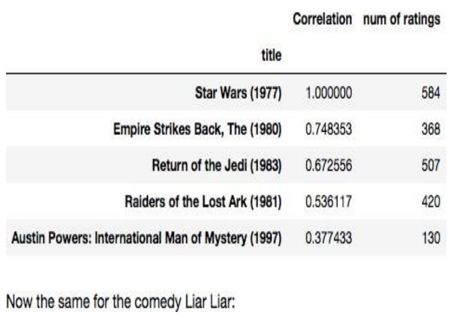
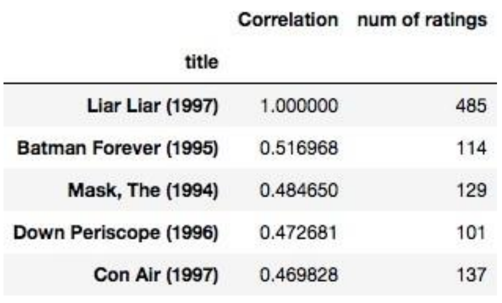


Figure 7.5.2: graphical representation of rating columns



title	Correlation	num of ratings
Star Wars (1977)	1.000000	584
Empire Strikes Back, The (1980)	0.748353	368
Return of the Jedi (1983)	0.672556	507
Raiders of the Lost Ark (1981)	0.536117	420
Austin Powers: International Man of Mystery (1997)	0.377433	130

Now the same for the comedy Liar Liar:



title	Correlation	num of ratings
Liar Liar (1997)	1.000000	485
Batman Forever (1995)	0.516968	114
Mask, The (1994)	0.484650	129
Down Periscope (1996)	0.472681	101
Con Air (1997)	0.469828	137

Table 7.6: similar movies of starwars and liarliar

Chapter 8

CONCLUSION AND FUTURE ENHANCEMENT

8.1 Conclusion

This recommendation system recommends different movies to user. Content-based recommendation systems are constrained to people, these systems don't prescribe things out of the box. These systems work on individual users’ ratings, hence limiting your choice to explore more. While our system which is based on a collaborative approach computes the connection between different clients and relying upon their ratings, prescribes movies to others who have similar tastes, subsequently allowing users to explore more. It is a web application that allows users to rate movies as well as recommends them appropriate movies based on other's ratings.

8.2 Future Enhancements

- In the proposed approach, it has considered similar 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, Amazon Prime, Hotstar and Netflix datasets and the performance can be computed in the future.
- Nowadays there are some more websites to watch movies so that we can avoid going to theatres, where it will be developed more in future.

Chapter 9

REFERENCES

[1] https://www.geeksforgeeks.com for datasets.

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

[3] Von Reischach, F., Guinard, D., Michahelles, F., & Fleisch, E. (2009, March). A mobile product recommendation system interacting with tagged products. 2009 IEEE international conference on pervasive computing and communications (pp. 1-6). IEEE.

[4] Dakhel, G. M., & Mahdavi, M. (2011, December). A new collaborative filtering algorithm using K-means clustering and neighbors' voting. 2011 11th International conference on hybrid intelligent systems (HIS) (pp. 179-184). IEEE.

[5] Cui, B. B. (2017). Design and implementation of movie recommendation system based on Knn collaborative filtering algorithm. ITM web of conferences (Vol. 12, p. 04008). EDP Sciences.