

**SAVITRIBAI PHULE PUNE UNIVERSITY**

**A PRELIMINARY PROJECT REPORT ON**

**“MEDIA RECOMMENDATION SYSTEM”**

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE

OF

**BACHELOR OF ENGINEERING**

**(COMPUTER ENGINEERING)**

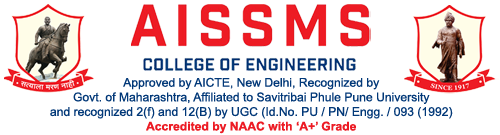
BY

TANMAY BHUSKUTE 18CS006

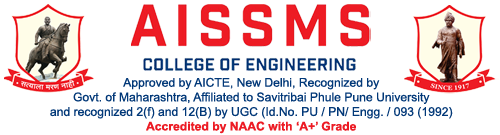
AMIT JEVE 18CS024

NIHAL SHAH 18CS049

TEJAS SHAH 18CS050



No. 1, Kennedy Road, Near RTO Office Sangamvadi, Shivajinagar, Pune - 411001



**CERTIFICATE**

This is to certify that the project report entitles

**“Media Recommendation System”**

Submitted by

TANMAY BHUSKUTE 18CS006

AMIT JEVE 18CS024  
NIHAL SHAH 18CS049

TEJAS SHAH 18CS050

is a bonafide work carried out by them under the supervision of Prof. B.A. Patil and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University Pune for the award of the degree of Bachelor of Engineering (Computer Engineering). This project work has not been earlier submitted to any other Institute or University for the award of any degree or diploma.

Prof. B. A. Patil Dr. D. P. Gaikwad Prof. S.R. Nalamwar  
 Project Guide H.O.D Computer Engineering Project Coordinator

Dr. D.S. Bormane  
Principal  
AISSMS College of Engineering

Place: Pune  
Date:

**ACKNOWLEDGEMENT**

We would like to extend my sincere gratitude and thanks to our guide Prof. B.A. Patil, for her invaluable guidance and for giving us useful inputs and encouragement time and again, which inspired us to work harder. Due to her forethought, appreciation of the work involved and continuous imparting of useful tips, this report has been successfully completed.

We are extremely grateful to Dr. D.P. Gaikwad, Head of the Department of Computer Engineering, for his encouragement during the course of the project work.

We also extend our heartfelt gratitude to the staff of Computer Engineering Department for their cooperation and support.

We also take this opportunity to thank all our classmates, friends and all those who have directly or indirectly provided their overwhelming support during our project work and the development of this report.

Tanmay Bhuskute

Amit Jeve

Nihal Shah

Tejas Shah

(B.E Computer Engineering.)

**ABSTRACT**

This project discusses about a single recommendation platform for movies, TV series, books and songs. A recommender system, or a recommendation system is a subclass of information filtering system that seeks to predict the "rating" or "preference" a user would give to an item.

It also helps users to find the movies of their choices based on the movie experience of other users in efficient and effective manner without wasting much time in useless browsing. Previous approaches in recommendation systems (RS) include Content-based-filtering and collaborative filtering. These approaches have certain limitations as like the necessity of the user history as they visit.

So as to overcome the effect of such dependencies, we intend to use a Hybrid Recommendation System. This system uses both Collaborative filtering & Content based filtering for recommending movies, TV series, books, songs.

In this way, the system performance will be greatly improved through the integration of these two.

**INDEX**

**LIST OF FIGURES & TABLES** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 06

**1. INTRODUCTION** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 08

1.1 BACKGROUND . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 08

1.2 PROBLEM STATEMENT . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 08

1.3 PURPOSE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 09

1.4 SCOPE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 09

**2. LITERATURE SURVEY** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10

2.1 PRESENT WORK . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10

2.2 PROPOSED WORK . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

**3. SOFTWARE REQUIREMENTS SPECIFICATION** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12

3.1 INTRODUCTION . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .12

3.1.1 PROJECT SCOPE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12

3.1.2 USER CLASSES AND CHARACTERISTICS . . . . . . . . . . . . . . . . . . . . . . . . . . . 12

3.1.3 OPERATING ENVIRONMENT . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12

3.1.4 DESIGN AND IMPLEMENTATION CONSTRAINTS . . . . . . . . . . . . . . . . . . . . 13

3.1.5 ASSUMPTIONS AND DEPENDENCIES . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13

3.2 EXTERNAL INTERFACE REQUIREMENTS . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14

3.2.1 USER INTERFACES . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14

3.2.2 HARDWARE INTERFACES . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14

3.2.3 SOFTWARE INTERFACES . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14

3.2.4 COMMUNICATION INTERFACES . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14

**4. SYSTEM ANALYSIS**. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

4.1 ANALYSIS MODEL . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

4.1.1 DATA FLOW DIAGRAMS . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

4.2 SYSTEM IMPLEMENTATION PLAN . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16

4.2.1 PROJECT PLAN AND SCHEDULE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16

4.2.2 PROJECT ESTIMATION . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17

4.2.3 RISK ANALYSIS . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18

4.2.4 SDLC PHASES . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18

4.2.5 GANTT CHART . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19

**5. SYSTEM DESIGN** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ….. . 20

5.1 SYSTEM ARCHITECTURE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21

5.2 UML DIAGRAMS . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21

5.2.1 USE CASE DIAGRAM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .21

5.2.2 ACTIVITY DIAGRAM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22

5.2.3 STATE CHART DIAGRAM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23

5.2.4 CLASS DIAGRAM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24

**6. PROJECT IMPLEMENTATION**. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26

6.1 IMPORTING DATASET FROM KAGGLE. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26

6.1.1 DATASETS. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .26

6.1.2 JUPYTER NOTEBOOK. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .26

6.1.3 TRAIN DATA. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .26

6.1.3.1 IMPORTING DATASET USING PANDAS. . . . . . . . . . . . . . . . . . . . . . . . .26

6.1.3.2 DATA CLEANING . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .26

6.1.4 FITTING DATA INTO MODELS. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26

**7. TECHNICAL SPECIFICATIONS** . . . . .. . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26

7.1 ADVANTAGES . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26

7.2 DISADVANTAGES . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27

7.3 APPLICATIONS . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27

**8. TESTING**. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29

**9. RESULTS**. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .29

**10. CONCLUSIONS** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29

**11. REFERENCES** . . . . . . . . . . . . . . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29

**APPENDIX**. . . . . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

PUBLICATION. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

**LIST OF FIGURES & TABLES**

Figure 4.1.1.1 – Data Flow Diagram . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

Figure 1.2 – Gantt Chart . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .19

Figure 1.3 System Architecture . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20

Figure 1.4 Use Case Diagram . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .21

Figure 1.5 Activity Diagram . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22

Figure 1.6 State Chart . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .23

Figure 1.7 Class Diagram . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24

Table 1.1 Project Plan and Schedule . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .16

Table 1.2 COCOMO Model . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .17

Table 1.3 Risk Analysis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18

Table 1.4 SDLC Phases . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .18

**CHAPTER 1**

**INTRODUCTION**

**1.1 BACKGROUND**

The explosive growth in the amount of available digital information and the number of visitors to the Internet have created a potential challenge of information overload which hinders timely access to items of interest on the Internet. This has increased the demand for recommender systems more than ever before. Recommender systems are information filtering systems that deal with the problem of information overload by filtering vital information fragment out of large amount of dynamically generated information according to user’s preferences, interest, or observed behavior about item. Recommender system has the ability to predict whether a particular user would prefer an item or not based on the user’s profile.

**What Are Recommender Systems?**

Recommender systems are so commonplace now that many of us use them without even knowing it. Because we can't possibly look through all the products or content on a website, a recommendation system plays an important role in helping us have a better user experience, while also exposing us to more inventory we might not discover otherwise.

Some examples of recommender systems in action include product recommendations on Amazon, Netflix suggestions for movies and TV shows in your feed, recommended videos on YouTube, music on Spotify, the Facebook newsfeed and Google Ads.

**1.2 PROBLEM STATEMENT**

In today's market there is no single recommender system available for all 3 major content sources of movies, books and songs. So, our idea is to develop a recommender system for movies, tv series, music and books on a webapp. The system should also recommend songs based on the movies watched and liked.

**1.3 PURPOSE**

**Why Do We Need Recommender Systems?**

We now live in what some call the “era of abundance”. For any given product, there are sometimes thousands of options to choose from. Think of the examples above: streaming videos, social networking, online shopping; the list goes on. Recommender systems help to personalize a platform and help the user find something they like.

The easiest and simplest way to do this is to recommend the most popular items. However, to really enhance the user experience through personalized recommendations, we need dedicated recommender systems.

From a business standpoint, the more relevant products a user finds on the platform, the higher their engagement. This often results in increased revenue for the platform itself. Various sources say that as much as 35–40% of tech giants’ revenue comes from recommendations alone.

Following are the reasons why recommendation system is used:

* They help the user find items of their interest
* Helps the item provider to deliver their items to the right user
  + To identify the most relevant products for each user
  + Showcase personalized content to each user
  + Suggest top offers and discounts to the right user
* Websites can improve user-engagement
* It increases revenues for business through increased consumption

**1.4 SCOPE**

After implementation, this project recommends movies, books and songs using machine learning algorithms. Using this recommendation system user will be able to find best movies, books and songs to watch and read.

After applying machine learning algorithms to the recommender system, we help users to find the content of their choices based on the experience of other users in efficient and effective manner without wasting much time in useless browsing.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 PRESENT WORK**

**Collaborative Filtering:** Collaborative filtering approaches build a model from a user's past behavior (items previously purchased or selected and/or numerical ratings given to those items) as well as similar decisions made by other users. e.g.: items previously purchased or selected in a marketplace.

**Disadvantages:**

* A collaborative filtering system does not necessarily succeed in automatically matching content to one's preferences. Unless the platform achieves unusually good diversity and independence of opinions, one point of view will always dominate another in a particular community.
* As in the personalized recommendation scenario, the introduction of new users or new items can cause the cold start problem, as there will be insufficient data on these new entries for the collaborative filtering to work accurately. In order to make appropriate recommendations for a new user, the system must first learn the user's preferences by analyzing past voting or rating activities. The collaborative filtering system requires a substantial number of users to rate a new item before that item can be recommended.

**Content Based Filtering:** Content-based filtering is a type of recommender system that attempts to guess what a user may like based on that user’s activity. Content-based filtering makes recommendations by using keywords and attributes assigned to objects in a database (e.g., items in an online marketplace) and matching them to a user profile.

**Disadvantages:**

* Whereas it needs very little information to start, but it is far more limited in scope (for example, it can only make recommendations that are similar to the original object).
* Since the feature representation of the items are hand-engineered to some extent, this technique requires a lot of domain knowledge. Therefore, the model can only be as good as the hand-engineered features.
* 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.

**2.2 PROPOSED WORK**

We propose a hybrid recommendation system to overcome the shortcomings of the existing systems.

Hybrid recommendation: By making content-based and collaborative-based predictions separately and then combining them; by adding content-based capabilities to a collaborative-based approach (and vice versa); or by unifying the approaches into one model e.g.: a good example of the use of hybrid recommender systems, The website makes recommendations by comparing the watching and searching habits of similar users (i.e., collaborative filtering) as well as by offering movies that share characteristics with films that a user has rated highly (content-based filtering) and finally we rank all these into a list with the best recommendation on top.

**CHAPTER 3**

**SOFTWARE REQUIREMENTS SPECIFICATION**

**3.1 INTRODUCTION**

**Why a Content Recommendation system?**

Today the amount of information in the internet growth very rapidly and people need some instruments to find and access appropriate information. One of such tools is called recommendation system. Recommendation systems help to navigate quickly and receive necessary information. The main aim of the Recommendation System is to give proper meaningful suggestion to person for specific items based on users’ mood and interest towards particular items, and provide good human computer interaction. It has seen a boom in recent years. Hybrid recommendation systems are mix of single recommendation systems as sub-components. This hybrid approach was introduced to cope with a problem of conventional recommendation systems such as “cold start”. The cold start problem occurs when the system is unable to form any relation between users and items for which it has insufficient data.

**3.1.1 PROJECT SCOPE**

After implementation, this project recommends movies, books and songs using machine learning algorithms. Using this recommendation system user will be able to find best movies, books and songs to watch and read. After applying machine learning algorithms to the recommender system, we help users to find the content of their choices based on the experience of other users in efficient and effective manner without wasting much time in useless browsing.

**3.1.2 USER CLASSES AND CHARACTERISTICS**

This project gives a recommendation which is faster, easier and concise. Consequently, the application will be able to predict the media using user’s and peer’s interest. And the user interface is as simple as possible.

Most importantly, the application must be reliable. In any case, the application must accurately predict the media and display it as soon as possible so that user does not have to waste time searching for the appropriate media.

**3.1.3 OPERATING ENVIRONMENT**

The main component of our project is the application that we are designing to ease the process of recommending media for a user. It is a web application along with internet connection requirement. The application will require to acquire significant amount of data from the internet so it requires a decent amount of storage space also it can be used over a network of computers so a standard configuration of the computers over network required. It will not require any cloud support since the whole system will work on an individual’s desktop-based operating system given that the basic storage requirements are met and we have a good internet connection. The Graphical User Interface (GUI) required for this software will be built using Python so we require an editor with Python support. Also, the algorithms that are going to be implemented in the system will be based on python Implementation. Beyond that, the application is a self-contained unit and will not rely on any other Desktop OS related software components except for storage space. This software application will be interacting with user at start to input the various features such as previously liked movie, songs, books. to gather data for prediction of media and at the end for displaying of recommendation. While the rest of the time the application will be running itself and there will be no interaction between software and user. The software application will operate on a network of computers having a internet connection and windows 7/8 or any other new versions as operating system with 4GB of RAM and minimum 1TB of allocated storage space per computer in network of computers.

**3.1.4 DESIGN AND IMPLEMENTATION CONSTRAINTS**

The primary design constraint is the Desktop platform. Since the application is designated for Desktop Systems, effective GUI and well user friendliness will be the major design considerations. Creating a user interface which is both effective and easily navigable is important. Also, as we are utilizing the database so storage space needs to be considered for smooth functioning of system. Other constraints such as memory and processing power are also worth considering. The analysis and prediction system are meant to be quick and responsive even when dealing with large amount of data so each feature of the software must be designed and implemented considering efficiency. As our system involves Machine Learning algorithm the system must consider the requirements of the algorithm for the format of input and output generated and their individual working efficiency and its contribution to overall software applications efficiency. The software will give the desired results only if the specified software requirements are satisfied.

At present only the excel file containing the data and the algorithm’s output format is considered. The system will accept details as input from the user and generates an output.

Application software designed must implement the algorithm effectively on the collected data and predict the expected result successfully also the interface of software must be easy and simple to be understood by the user and no extra efforts needed by them to understand the usage of software.

**3.1.5 ASSUMPTIONS AND DEPENDENCIES**

This project will work on the minimum system specifications as follow:

Any windows system.

**3.2 EXTERNAL INTERFACE REQUIREMENTS**

**3.2.1 USER INTERFACES**

For our system human computer interaction is done through Graphical User Interface. The user interface window contains a Web page that asks the user to give ratings to the media explored. The system accepts these values from user and start its functioning.

**3.2.2 HARDWARE INTERFACES**

The system has following hardware requirements or interfaces:

Keyboard: To enter media ratings.

Display Screen: To display web page and also to display a value after it recommends media.

**3.2.3 SOFTWARE INTERFACES**

The software interfaces will be the application software developed which runs on windows operating system. Our web application takes input i.e., Ratings of the media from user and starts its functioning. It gives the recommendations to the user.

**3.2.4 COMMUNICATION INTERFACES**

Graphical User Interface is the communication interface between user and the system.

**CHAPTER 4**

**SYSTEM ANALYSIS**

**4.1 ANALYSIS MODEL**

**4.1.1 DATA FLOW DIAGRAMS**

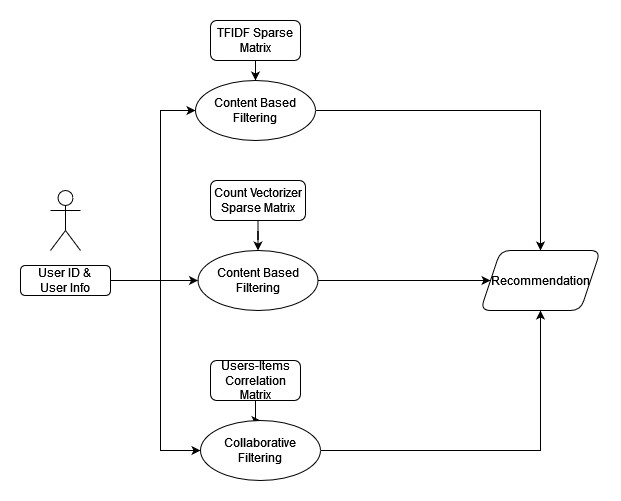


Figure 1.1 – Data Flow Diagram

**4.2 SYSTEM IMPLEMENTATION PLAN**

**4.2.1 PROJECT PLAN AND SCHEDULE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr no | Task Name | Start Date | Finish Date | Status |
| 1 | Topic Selection | Sept 6,2021 | Sept 29,2021 | Completed |
| 2 | Abstract | Sept 30, 2021 | Oct 2, 2021 | Completed |
| 3 | Requirement  Gathering | Oct 3, 2021 | Oct 5, 2021 | Completed |
| 4 | Distribution of Roles and  Responsibilities | Oct 6, 2021 | Oct 8, 2021 | Completed |
| 5 | Project Planning | Oct 9, 2021 | Oct 13, 2021 | Completed |
| 6 | Synopsis | Oct 15, 2021 | Oct 17, 2021 | Completed |
| 7 | System  Architecture | Oct 18, 2021 | Oct 19, 2021 | Completed |
| 8 | Review 1 and  Review 2 | Oct 20, 2021 | Dec 7, 2021 | Completed |
| 9 | Review 3 and  Review 4 | Dec 8, 2021 | Dec 15, 2021 | Complete |
| 10 | Final PPT | Dec 16, 2021 | Dec 18, 2021 | Complete |
| 11 | Report Generation | Dec 20, 2021 | Jan 6, 2021 | Complete |

**4.2.2 PROJECT ESTIMATION**

**Project class:**

We have determined our project fits the characteristics of Semi-detached mode as project at college level and is an intermediate (in size and complexity) software project in which teams with mixed experience levels must meet a mix of rigid and less than rigid requirements.

Number of Code Lines:

We estimate our project will have 10000 Delivered Source instructions. So, the Basic COCOMO model equations are as follows:

The basic COCOMO equations take the form.

E = ab(KLOC)bb

D = cb(E)db

P = E/D

where

E is the effort applied in person-months,

D is the development time in chronological months,

KLOC is the estimated number of delivered lines of code for the project (express in thousands).

The coefficients ab, bb, cb and db are given in the following form.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Software project | ab | bb | Cb | Db |
| Organic | 2.4 | 1.05 | 2.5 | 0.38 |
| Semi-detached | 3.0 | 1.12 | 2.5 | 0.35 |
| Embedded | 3.6 | 1.20 | 2.5 | 0.32 |

Table 3.4.2 COCOMO Model

Calculations:

So, this project comes under semi-detached mode,

Efforts Applied = E = 3.0 \* (10)1:1 = 39.54 person-months

Development Time = D = 2.5 \* (39:54)0:35 = 9.055 months

People Required = P = 39.54/9. 055 = 4.366 count

**4.2.3 RISK ANALYSIS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr No. | Risk | Risk Type | Probability | Impact | Mitigation Plan |
| 1 | Cold Start | Technical | High | High | Use Hybrid Recommendation |
| 2 | Very Huge Database | Technical | Medium | Medium | Making large storage space available. |
| 3 | Updating of Database | Technical | High | Medium | Update Database on timely basis. |
| 4 | Delay in schedule | Operational | High | High | Do the right work in right  time. |
| 5 | Team Management: Conflicts of Opinions and  working styles. | Management | Medium | Medium | To respect and be open minded about others  working styles. |

**4.2.4 SDLC PHASES**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr No | Task Name | Start Date | Finish Date | Duration (Days) | Complete (%) |
| 1 | Requirement  Gathering | Sept 30,21 | Oct 20,21 | 20 | 100 |
| 2 | Design and  Modeling | Oct 1,21 | Nov 10,21 | 40 | 100 |
| 3 | Implementation | Dec 20,21 | Feb 8,22 | 50 | 100 |
| 4 | Testing | Feb 9,22 | Feb  22,22 | 13 | 100 |
| 5 | Deployment | Feb 23,22 | Mar 25,22 | 30 | 100 |

**4.2.5 GANTT CHART**

Figure 1.2 – Gantt Chart

**CHAPTER 5**

**SYSTEM DESIGN**

**5.1 SYSTEM ARCHITECTURE**

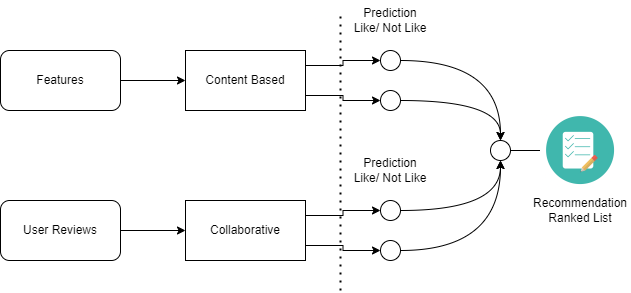


Figure 1.3 – System Architecture

System architecture describes an overall outline of the software system and the relationships, constraints, and boundaries between components. It provides a basic view of the physical deployment of the software system and its evolution roadmap. The above diagram shows the system architecture of a Media Recommendation system using both Content & Collaborative based filtering techniques (Hybrid Model).

**5.2 UML DIAGRAMS**

**5.2.1 USE CASE DIAGRAM**



Figure 1.4 – Use Case Diagram

A use case diagram is a graphical depiction of a person's possible interactions with a machine. A use case diagram suggests various use instances and distinct types of users the system has and could frequently be accompanied by way of different styles of diagrams as nicely. The use cases are represented with the aid of either circles or ellipses.

**5.2.2 ACTIVITY DIAGRAM**

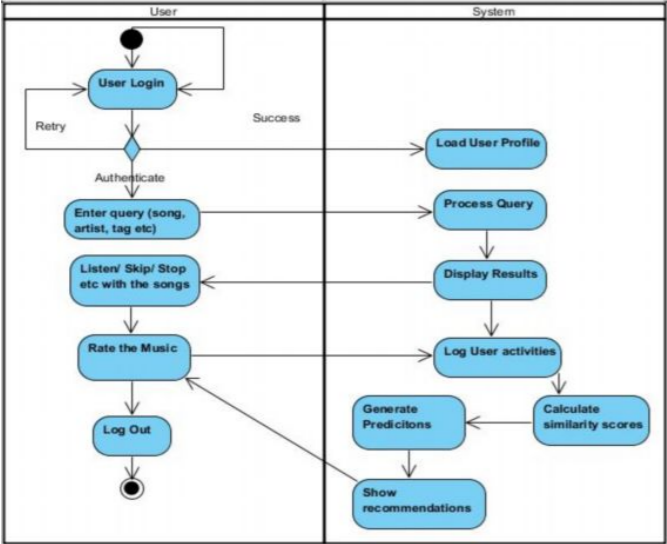
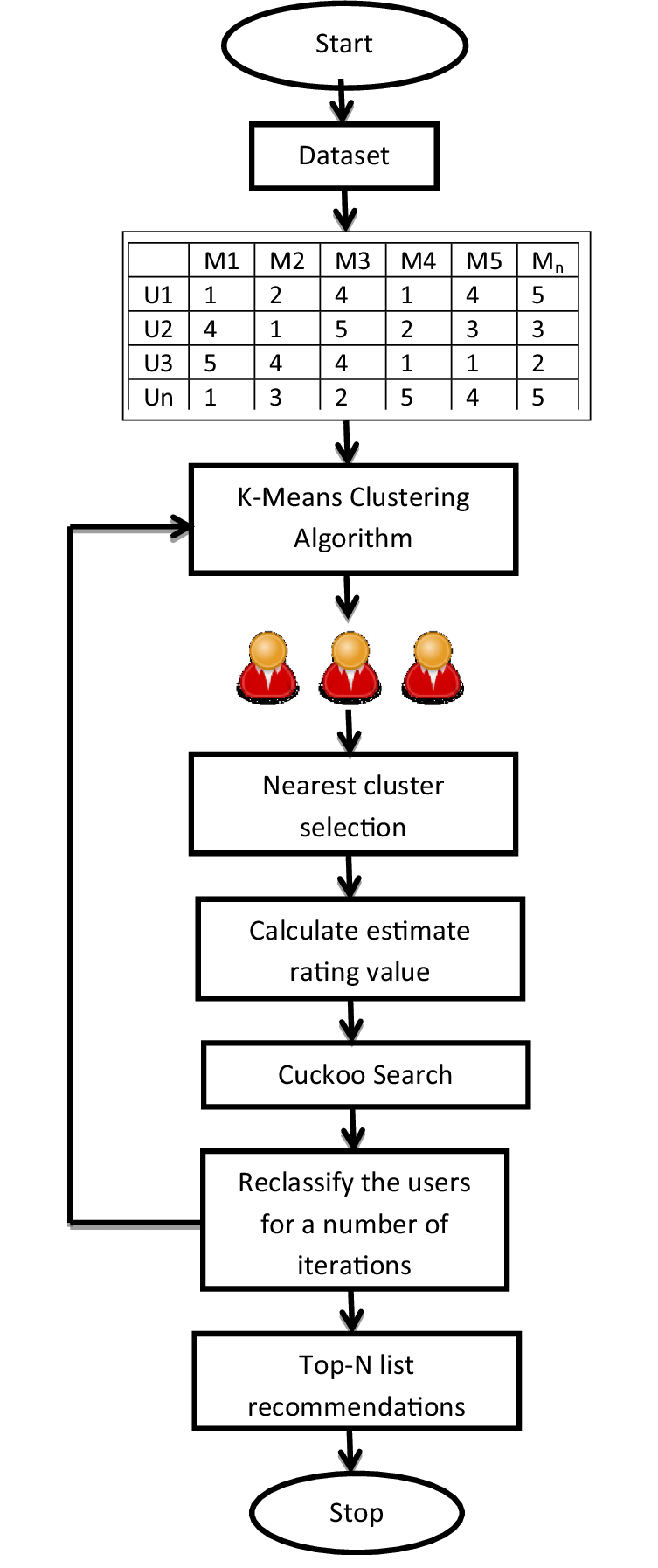


Figure 1.5 – Activity Diagram

An activity diagram is a behavioral diagram which depicts the behavior of a system. An activity diagram portrays the manage glide from a begin factor to a finish factor showing the numerous decision paths that exist while the interest is being executed

**5.2.3 STATE CHART DIAGRAM**

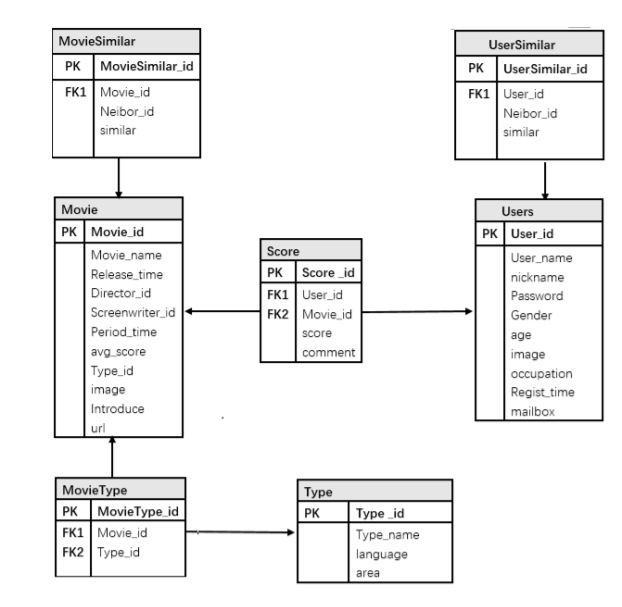
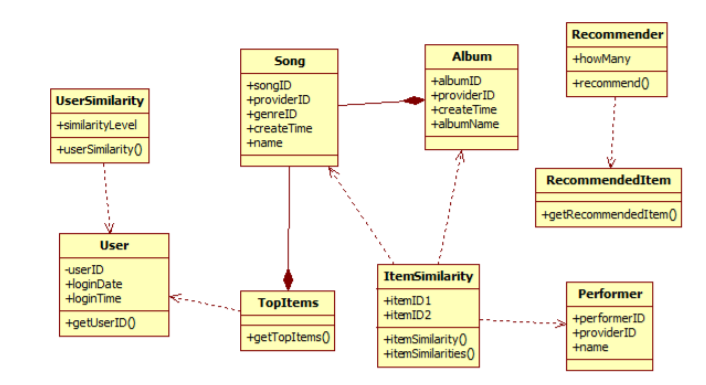


Cosine Similarity, Pearson co-relation

Figure 1.6 – State Chart Diagram

A state diagram is the graphical illustration of a single object in response to a series of events in a system. State diagrams display a behavioral version along with states, state transitions and moves. State diagrams depict the permitted states and transitions as well as the events that impact those transitions.

**5.2.4 CLASS DIAGRAM**

  
 Figure 1.7 – Class Diagrams

A Class diagram models static view of an application. Class diagram is not best used for visualizing, describing, and documenting specific elements of a device but additionally for constructing executable code of the software program application. Class diagram describes the attributes and operations of a class and additionally the restrictions imposed on the system. The class diagrams are widely used inside the modelling of object-oriented systems due to the fact they may be the simplest UML diagrams, which may be mapped immediately with different object-oriented languages. Class diagram indicates a set of classes, interfaces, associations, collaborations, and constraints. It’s also known as a structural diagram.

**CHAPTER 6**

**PROJECT IMPLEMENTATION**

**6.1 IMPORTING DATASET FROM KAGGLE**

**WHAT IS KAGGLE?**

“**Kaggle** allows users to:

* + 1. Find and publish data sets
    2. Explore and build models in a web-based data-science environment
    3. Work with other data scientists and machine learning engineers
    4. **Enter competitions** to solve data science challenges.”

“Inside Kaggle you‟ll find all the code & data you need to do your data science work. Use over 19,000 public and 200,000 public Notebooks to conquer any analysis in no time.”

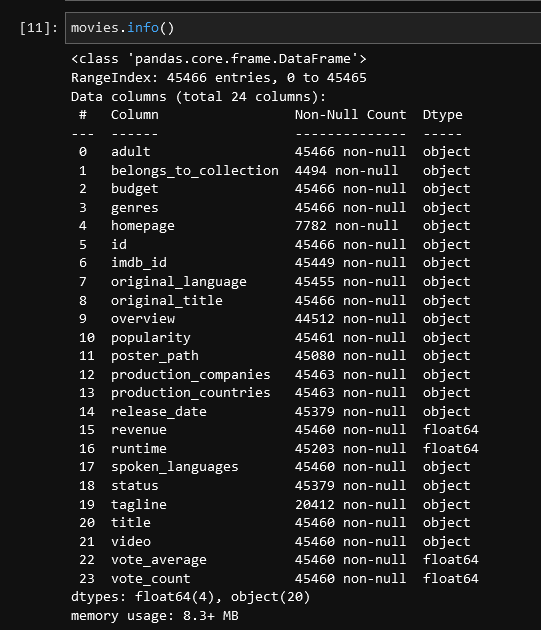
**6.1.1 DATASETS**

A dataset is a range of contiguous cells on an Excel worksheet containing data to analyze.

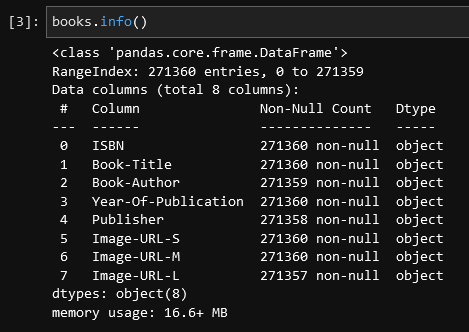
When arranging data on an Excel worksheet you must follow a few simple rules so that Analyze-it works with your data:

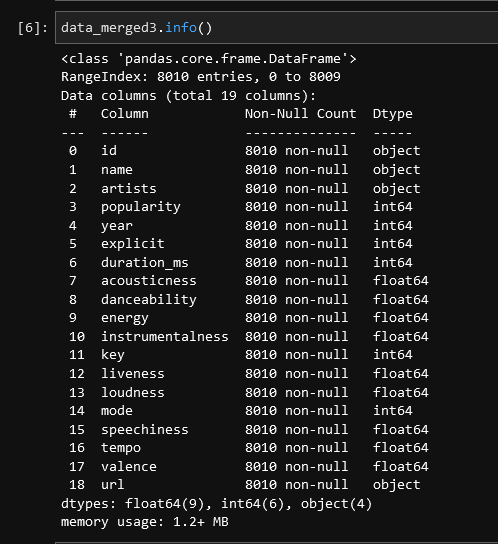
* + - 1. Title to clearly describe the data. If you do not specify a title, the cell range of the dataset (such as A3:C13) is used to refer to the dataset.
      2. A header row containing variable labels. Each variable name should be unique. Units of measurement can be included in the label by enclosing them in brackets after the name.
      3. Rows containing the data for each case. The number of rows is only limited by Excel (currently over one million).
      4. Columns containing the data for each variable.
      5. Optional: Labels in the first column to provide a meaningful name/identifier for each case.

When you use an Analyze-it command, the extent of a dataset is determined by scanning outwards from the active cell to include all surrounding contiguous cells. The extent is known when a blank row or column surrounding the dataset, or the edge of the worksheet, is reached.

****

**Fig. Movie Dataset**

****

**Fig. Books Dataset**

**Fig. Music Dataset**

**6.1.2 JUPYTER NOTEBOOK**

IPython notebook was developed by Fernando Perez as a web based front end to IPython kernel. As an effort to make an integrated interactive computing environment for multiple language, Notebook project was shifted under Project Jupyter providing front end for programming environments Juila and R in addition to Python.

A notebook document consists of rich text elements with HTML formatted text, figures, mathematical equations etc. The notebook is also an executable document consisting of code blocks in Python or other supporting languages.

Jupyter notebook is a client-server application. The application starts the server on local machine and opens the notebook interface in web browser where it can be edited and run from. The notebook is saved as ipynb file and can be exported as html, pdf and LaTex files.

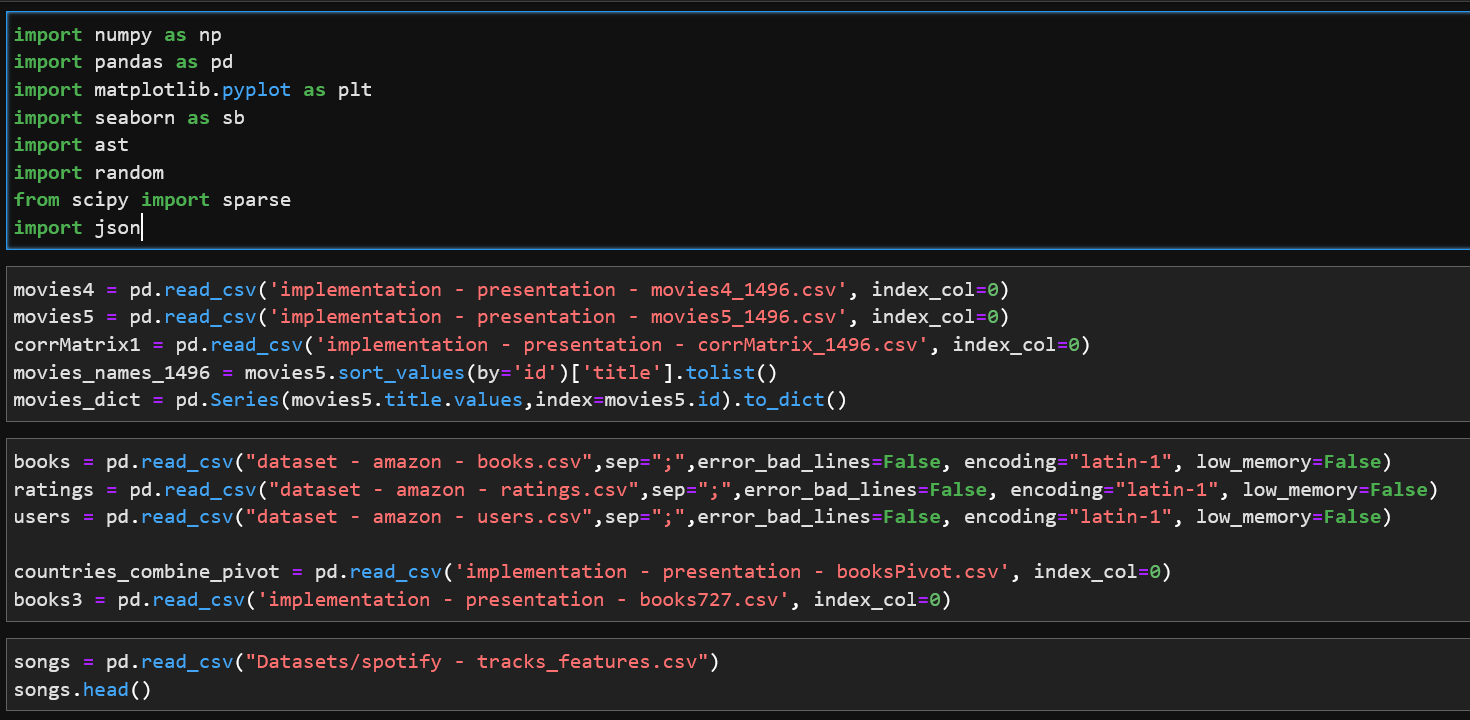
**6.1.3 TRAIN DATA**

**6.1.3.1 IMPORTING DATASET USING PANDAS**

You can easily import an Excel file into Python using [*Pandas*](https://datatofish.com/create-pandas-dataframe/). In order to accomplish this goal, you’ll need to use *read\_csv*.

In this short guide, you’ll see the steps to import an Excel file into Python using a simple example.

But before we start, here is a template that you may use in Python to import your Excel file:



**Fig. Import Dataset**

**6.1.3.2 DATA CLEANING**

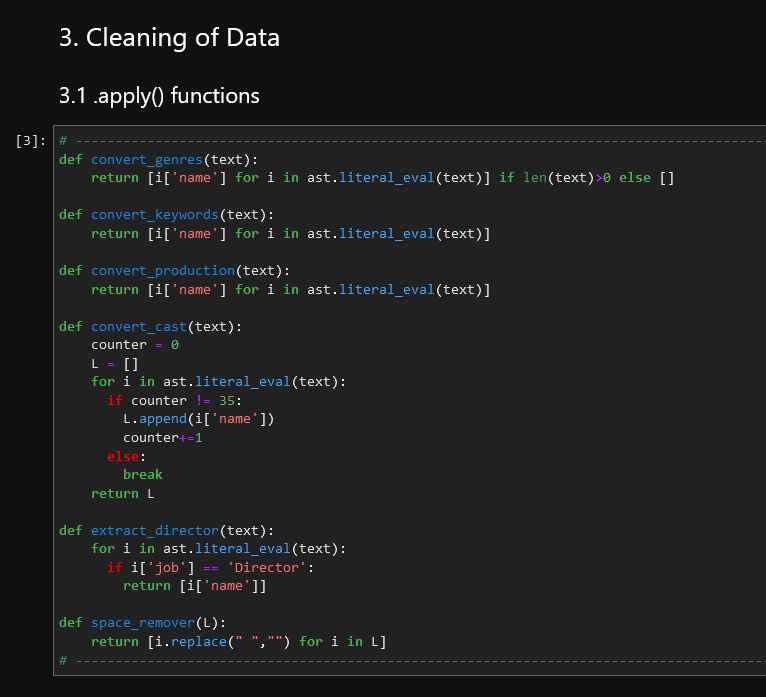
Prior to performing any sort of data analysis, data scientists spend a lot of time cleaning the dataset to get it in the refined form it needs to be for various data science techniques to be performed on it.

It is an important part of a data scientist’s job to handle messy data including missing values,

inconsistent formatting, and data types.

The process of data cleaning is roughly covered in the following steps:

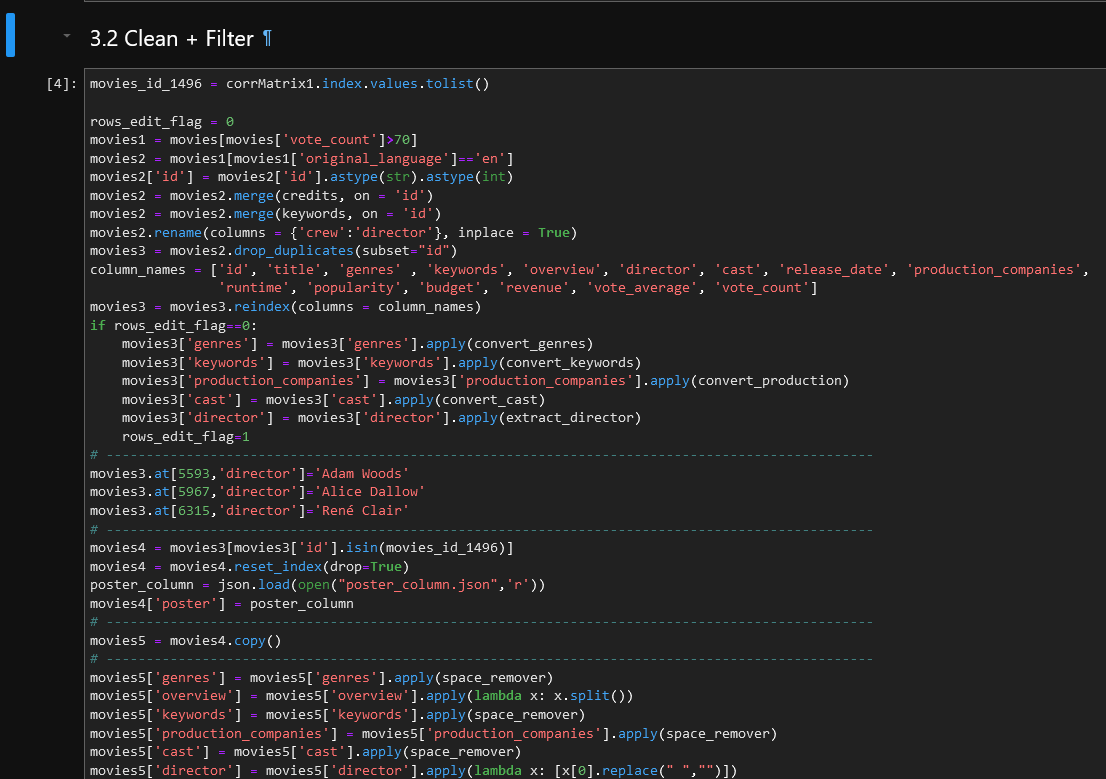
* + - Dropping irrelevant columns.
    - Renaming column names to meaningful names.
    - Making data values consistent.
    - Imputing missing values.



**Fig. Movies dataset preprocessing and cleaning**

****

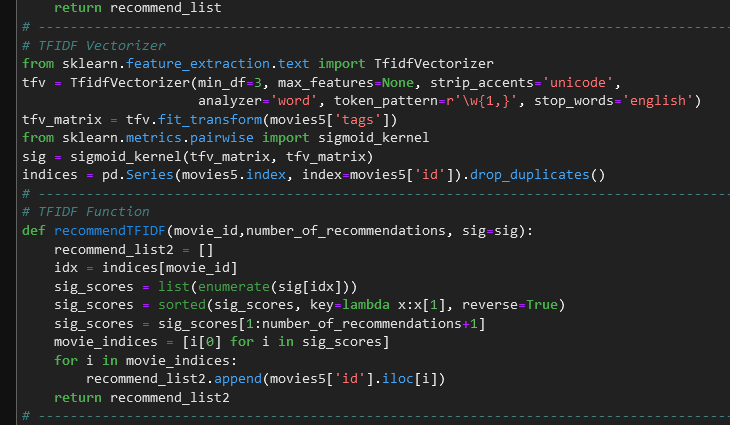
**Fig. Books dataset preprocessing and cleaning**

****

**Fig. Songs dataset preprocessing and cleaning**

**6.1.4 FITTING DATA INTO MODELS**

****

****

**CHAPTER 7**

**TECHNICAL SPECIFICATION**

**7.1 ADVANTAGES**

**7.1.1 Collaborative Filtering:**

- Domain knowledge is not required as the embedding is automatically learned.

- The model can help the user explore different types of media or different genres other than the user’s interests.

- The model only needs ratings of user with title of media matrix for a start point.

**7.1.2 Content Based Filtering:**

- The model doesn't need any data about other users, since the recommendations are specific to this user. This makes it easier to scale to a large number of users.

- The model can apprehend the specific interests of a user, and can recommend niche items that very few other users are interested in.

**7.1.3 Hybrid Based Filtering:**

- Higher performance, accuracy than filter.

- Good computational complexity.

- More flexible and robust for high dimensional data.

**7.1.4 Project Based:**

- This media system provides a better experience for the users by giving them a vast exposure to many different media they might be interested in.

- This recommender system combines two recommendation techniques to gain better performance with fewer of the drawbacks of any individual one.

- This system has a unique recommendation technique to recommend songs based on the movies you’ve watched.

- Helps users find the content of their choices based on the experience of different users in efficient and effective manner without wasting much time in useless browsing.

**7.2 DISADVANTAGES**

**7.2.1 Collaborative Filtering:**

- Scalability: Cannot handle fresh items.

- Hard to include side features for query/item.

- Cold Start Problem: If there is insufficient amount of reviews/ data available in form of ratings so the model cannot run the collaborative filtering algorithm. This is one of the major disadvantage of collaborative filtering.

**7.2.2 Content Based Filtering:**

- Since the feature representation of the items are hand-engineered to some extent, this technique requires a lot of domain knowledge.

- The model can only make recommendations based on existing interests of the user.

**7.2.3 Hybrid Based Filtering:**

- Classifier specific methods

- Dependents of the combination of different feature selection method

**7.2.4 Project Based:**

- People don’t have a tendency to rate movies, books after watching.

- Need an enormous dataset to start recommending appropriate media to users.

- Ever-changing datasets.

**7.3 APPLICATIONS**

- This media system provides a better experience for the users by giving them a vast exposure to many different media they might be interested in. The different media which the system is going to recommend are movies, songs, books.

- This recommender system combines two recommendation techniques to gain better performance with fewer of the drawbacks of any individual one.

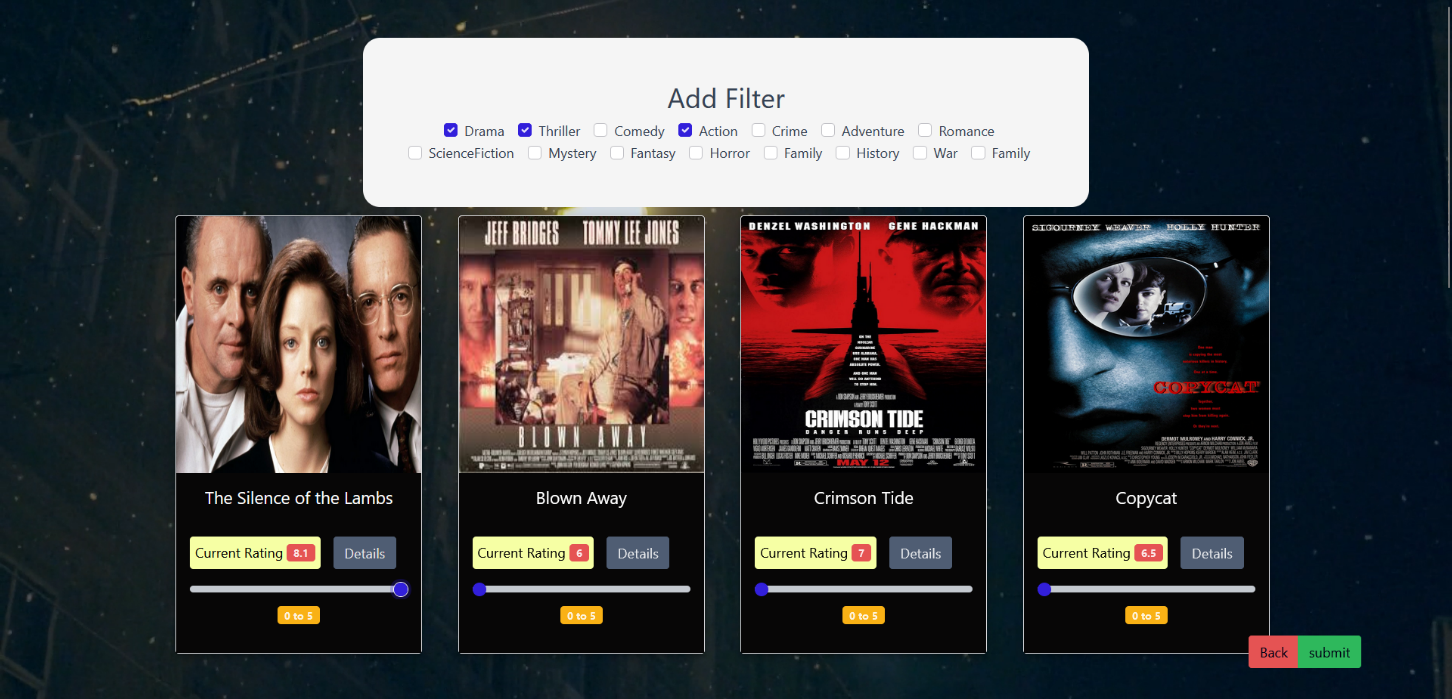
- This system has a unique recommendation technique to recommend songs based on the movies you’ve watched.

- Helps users to find the content of their choices based on the experience of other users in efficient and effective manner without wasting much time in useless browsing.

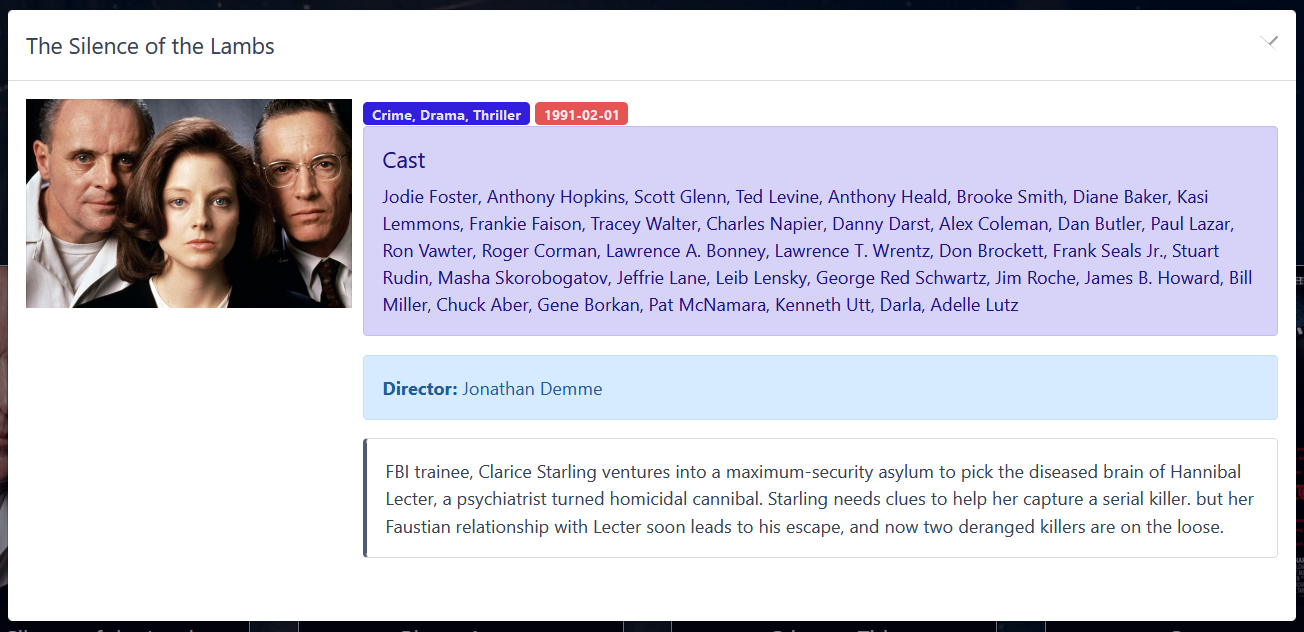
**CHAPTER 8**

**TESTING**

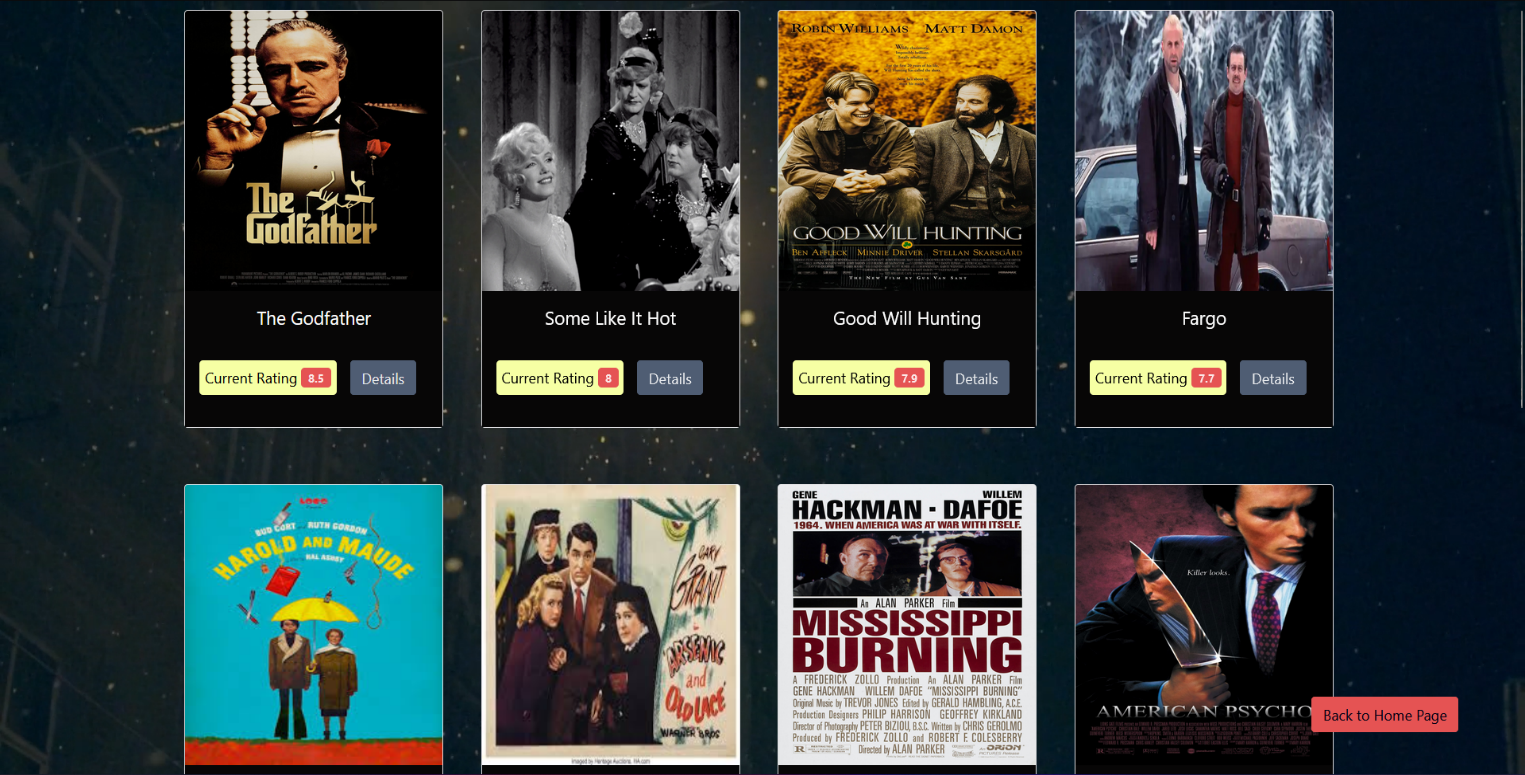
1. **MOVIES**
2. We have added a filter in this movie recommender, using this we have filtered different genres of movies like: drama, thriller, action. From this we gave five-star rating to a particular movie “The Silence of the Lambs”.



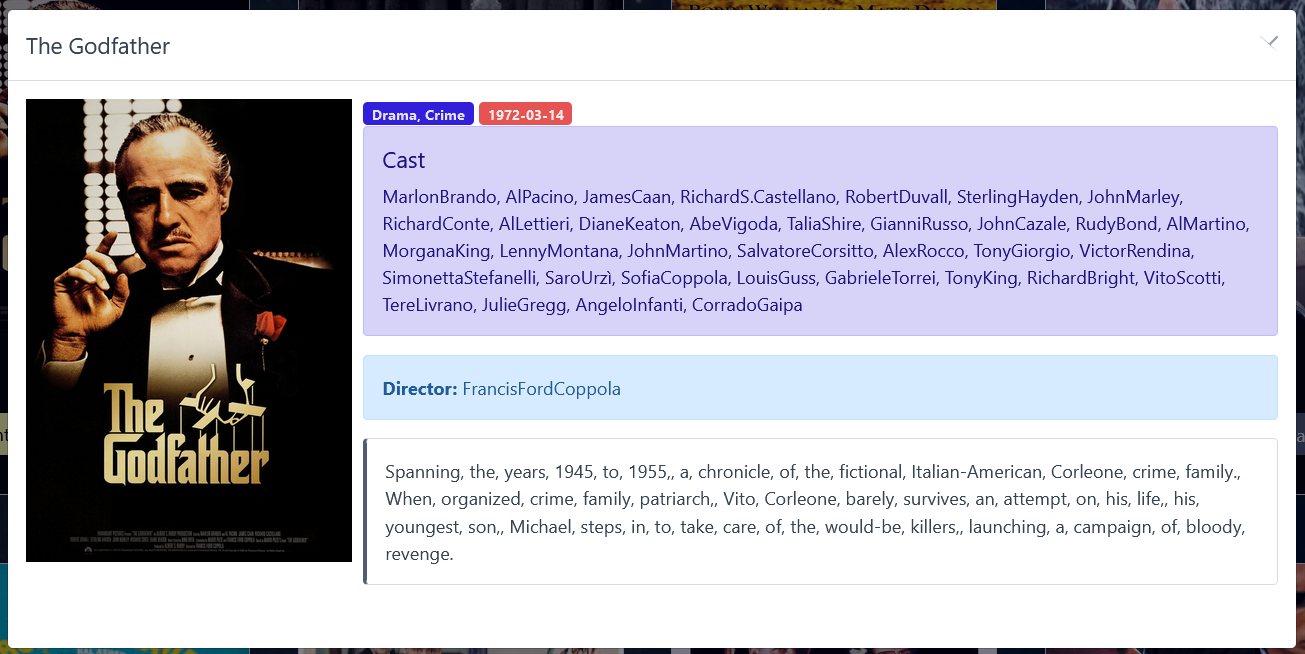
1. Here we click on the card of the movie and get all the details about it like: Cast, Release Date, Director and also the synopsis of the movie.

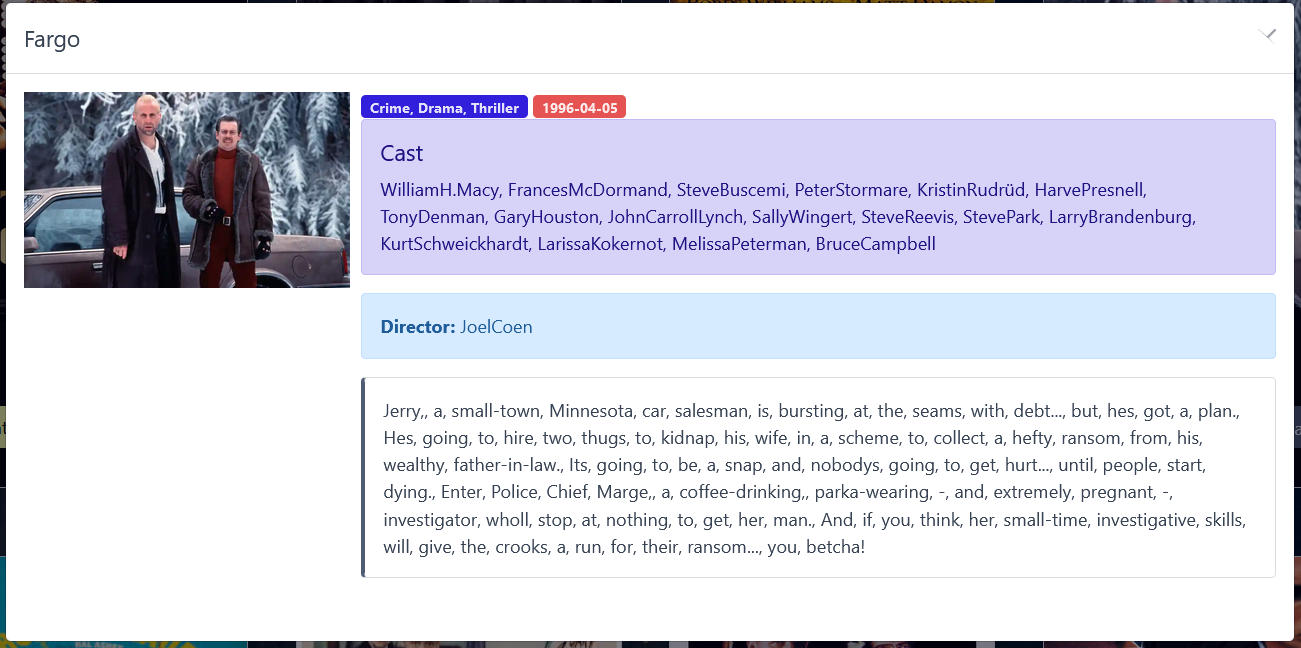


1. After we submit the movie, our algorithm then predicts or recommends us the movies similar to our selected movie and also of the same genre. Here we get movies like “The Godfather”, “Fargo” and “Good Will Hunting”.



1. To check the accuracy of our algorithm we can further check the details of movies recommended to us, whether they match or are in fact similar to the movie we selected earlier. After checking the details of The Godfather and Fargo we understand that the movies are similar.





**Chapter 9**

**RESULTS**

**9.1 EXPECTED RESULTS**

The user has to first login to account in the web-app. Then the user has to search for names/ genres of media which he/she already like and rate it. Then the system will recommend the user different media recommendation according to the likes of the user. The media recommendation system will be based on hybrid filtering recommendation system.

**Chapter 10**

**CONCLUSION**

Recommender systems open new opportunities of retrieving personalized information on the Internet. We come up with a strategy that focuses on dealing with user’s personal interests and based on his previous reviews, media are recommended to users. This strategy helps in improving accuracy of the recommendations. A personal profile is created for each user, where each user has access to his own history, his likes, ratings. It also helps in collecting authentic data with improved accuracy and makes the system more responsive.

**Chapter 11**

**REFERENCES**

1. *Cui, Bei-Bei. (2017). Design and Implementation of Movie Recommendation System Based on Knn Collaborative Filtering Algorithm. ITM Web of Conferences. 12. 04008. 10.1051/itmconf/20171204008.*
2. *C. M. Wu, D. Garg and U. Bhandary, "Movie Recommendation System Using Collaborative Filtering," 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), 2018, pp. 11-15, doi: 10.1109/ICSESS.2018.8663822.*
3. *Kashyap, A., Sneha Srivastava and Anup Jung Shah. “A Movie Recommender System: MOVREC using Machine Learning Techniques.” (2020).*
4. *Panwala, Kalp. (2020). "Various Types of Movies Recommendation System". 10.13140/RG.2.2.29566.41283.*
5. *W. Wenzhen, "Personalized Music Recommendation Algorithm Based on Hybrid Collaborative Filtering Technology," 2019 International Conference on Smart Grid and Electrical Automation (ICSGEA), 2019, pp. 280-283, doi: 10.1109/ICSGEA.2019.00071.L*
6. *Ifada, Noor & Rahman, Triyani & Sophan, Mochammad. (2020).Comparing Collaborative Filtering and Hybrid based Approaches for Movie Recommendation. 219-223. 10.1109/ITIS50118.2020.9321014.*
7. *S. Agrawal and P. Jain, "An improved approach for movie recommendation system," 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2017, pp. 336-342, doi: 10.1109/I-SMAC.2017.8058367.*
8. *Muppana, Mahesh & Kanmani, R. & B, Surendiran. (2020). Analysis of Movie Recommendation Systems; with and without considering the low rated movies. 1-4. 10.1109/ic-ETITE47903.2020.453.*
9. *M. Sunitha, T. Adilakshmi, Comparison Of User-Based Collaborative Filtering Model For Music Recommendation System With Various Proximity Measures, International Journal Of Innovative Technology And Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-6S2, April 2019.*
10. *Awawdeh, S.; Edinat, A.; Sleit, A. An Enhanced K-Means Clustering Algorithm for Multi-Attributes Data. Int. J. Comput. Sci. Inf. Secur. 2019, 17. Available online: https://scholar.google.com/scholar?hl=en&as\_sdt=0%2C5&q=An+Enhanced+K-means+ Clustering+Algorithm+for+Multi-+attributes+Data&btnG= (accessed on 14 April 2021).*

**APPENDIX**

**Publication**

*Tanmay Bhuskute, Amit Jeve, Nihal Shah, Tejas Shah, B. A. Patil, “MediaRec: A Hybrid Media Recommender System”, International Journal for Research in Applied Science & Engineering Technology (IJRASET), ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538, Volume 10 Issue V May 2022- Available at* [*www.ijraset.com*](http://www.ijraset.com)