Movies Recommendation System

A

PROJECT REPORT

Submitted in the partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING

Under the supervision

Of

Dr. Hemraj Saini

By

Manish Sharma(161377)

Aabhas Kashyap(161370)

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY,WAKNAGHAT

SOLAN -173234

HIMACHAL PRADESH,INDIA

MAY-2020

Student's Declaration

I hereby declare that the work presented in this report entitled **"Movie Recommendation System"** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** submitted in the department of Computer Science & Engineering and Information Technology. Jaypee University of Information Technology Waknaghat, Solan is an authentic record of my own work carried out over a period from January 2020 to May 2020 under the supervision of **Dr.Hemraj Saini** (Associate Professor, Department of CSE & IT).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student Signature)

Manish Sharma (161377)

Aabhas Kashyap (161370)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(HemRaj Saini)

(Supervisor Signature)

Supervisor Name: Dr. Hemraj Saini

Designation: Associate Professor

Department Name: Department of Computer Science & Engineering

Dated: 23/May/2020

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success. I am grateful to my project guide **Dr. Hemraj Saini** for the guidance, inspiration and constructive suggestions that helped me in the preparation of the project.

I also thank our colleagues who have helped me in successful completion of the project.

Dated : 23/05/2020

MANISH SHARMA(161377)

AABHAS KASHYAP(161370)

TABLE OF CONTENTS

ABBREVIATIONS

CF .... COLLABORATIVE FILTERING

CB .... CONTENT BASED

IB ....ITEM BASED

AI .... ARTIFICIAL INTELLIGENCE

UB… .... USER BASED

MF...MATRIX FACTORISATION

MAE..MEAN ABSOLUTE ERROR

LIST OF FIGURES

ABSTRACT

Presently a-days recommender systems are utilized in our everyday life. However, they are a long way from flawlessness.

In this venture we assemble a recommendation motor that adds an entirely different measurement to the film watching experience by giving ongoing customized motion picture recommendations to clients. We start with examining and comprehend the different sorts of recommendation systems and how they work. We initiate by creating and looking at the various types of models on a littler dataset of barely any appraisals. We build up the system utilizing distinctive Machine learning Algorithms, for example, Content Based Algorithms and Collaborative Filtering techniques.

We will attempt to build up an adaptable model to perform insights. At that point, we attempt to measure the system with the goal that it can deal with 200 evaluations by utilizing MS SQL server. We come to realize that for a compact dataset, executing client based community separating results with better and efficient yields.

Hence, our system adopts a collective long range informal communication strategy where a client's very own preferences are blended in with that of the whole network to produce important outcomes. Most existing film administrations don't customize their recommendations yet basically give a general rating to a motion picture. This essentially diminishes the estimation of every recommendation as it doesn't take into account the individual film inclinations of the client.

We attempt to assemble a recommendation system that offers a rundown of film proposals based on past client appraisals. The recommendation system is planned so that the client doesn’t not have to look for films however to find them through our recommendation procedure.

Additionally the clients get the opportunity to rate films they have seen. This information is then broke down, and recommendations are then come back to the client.

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

A recommendation system is nothing but a software which generally takes out the filtered information and provide suggestions with respect to movies, items etc for the users or clients. In our daily life we see recommendations everywhere. For example if we visit on amazon ,flipkart like sites we found a lot of recommendations at their homepage and these recommendations are somewhere related to us or we can just say that recommendation makes our work very easy in today's world.

Nowadays every person has variety of options or choices such as What to wear? Where to go? What to purchase? What not to purchase? and so on and some of the time user doesn't have a clue what to do? For example if you want buy a pair of shoe than there are many sites available for you on internet as per your search and it takes a lot of time to find out which one is best or for what you are looking for. At that time recommendation system plays a very important or crucial role in today's world. It exactly provides the content for which a user is generally looking for. Hence a recommendation system basically provides suggestions to the user according to his taste

But it is not as easy as it seems to be. In order to make a recommendation system one should know about the proper methodologies, calculations, strategies, algorithms etc. To make a recommendation system for different purposes is not an easy job as it requires different algorithms, calculations etc. in different areas such as if we want to make a movie recommendation system and another one is article recommendation system then in both the recommendation system we use different methodologies, algorithms and strategies. Hence in order to make a recommendation system for a purpose one should properly know about algorithms, methodologies etc. to make a system which is more reliable.

The two fundamental methodologies on which most of recommendation system works are:

- Content based recommendation systems.
- Collaborative recommendation systems.

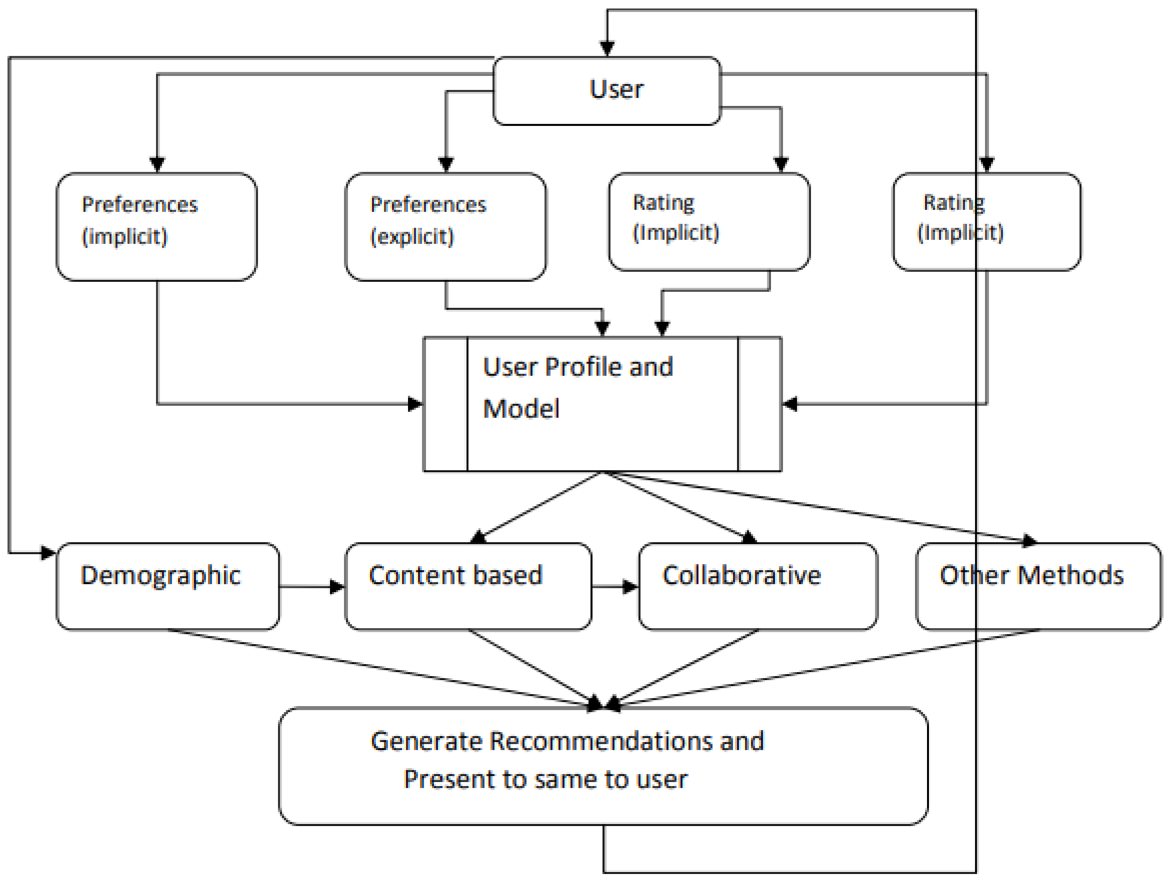

Figure1.1

The above figure shows the basic architecture or workflow of a recommendation system. A recommendation system generally filters the data from the extracted data that is collected by the system from the user either explicitly or implicitly and based on that provide recommendations to the user by using some similarity measures approaches or algorithms. Popular similarity measure approaches are:

- Cosine similarity
- Eucledian distance
- Dot product etc.

We discuss later on all the similarity measures and their pros and cons in further discussions.

1.1.1 Why recommendation system :

As we know that a recommendation system is generally a software which generally gives us filtered information, with respect to his/her taste or preferences of a customer or user.

- One of the main reason to develop a recommendation system is that in today's people have variety of information or options for a particular and this is due to the prevalence of internet. In past (about 20 year ago) there is not so much options available for the peoples. For example if a person go to the store at that time in order to purchase cd/dvd of blockbuster movie then he/she have limited option or you can say that it all depends on the size of the store at that time . But now a customer or user have unlimited choices over internet. For example in today's world if a user want to watch a movie then he/she have large number of choices and it takes a lot of time to find out a good thing on internet for which an user is actually looking for. At that time recommendation system , comes to play an important role in this world.

- It also saves customer or user time by recommending desired items or anything to the user.

- As it also helps for e-commerce sites by increase their overall profit just by providing good recommendations.

- It can also reduce employee workload of advertising their products as it helps customer to get their demands without exploring it on different sites.

So these are the some main reason to develop good recommendation system in today's world for the users/ client.

1.1.2 CONTENT BASE RECOMMENDATION SYSTEM

The main idea of content base recommendation is to collect the data about user or client and by using that data it provides recommendation, in another way we can say that a content base recommendation system generally analyze the user's profile and by applying some sort of calculations , algorithms etc. it provides recommendations to the user.

The data can be collected as explicitly or implicitly. For example if a user watches a movie and provide some rating to that movie than it is an explicit data and if a user just click on similar type of links than it is an implicit data. A good recommendation system filters out that data and provide recommendations.

A content base recommendation system just analyze a single user's profile and provides recommendation on his previous watch history or something else. Hence it works on single user profile.

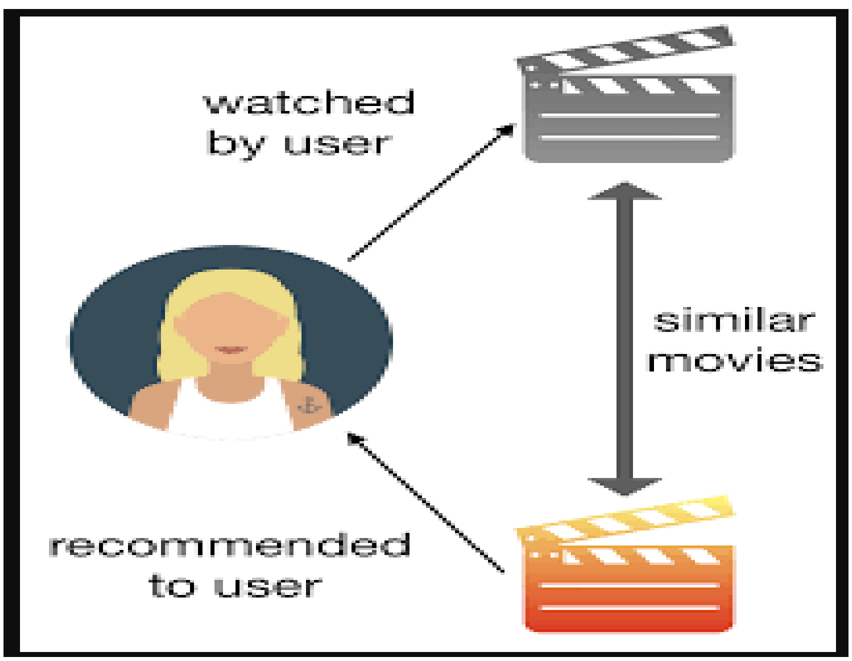

Figure 1.1

For example if a user watches movies than a user profile of his data is automatically store in database which generally predicts or filter out the information of the user behavior by using some algorithms and provide better recommendations to the user as shown in above figure 1.1.

Workflow of a content based recommendation:

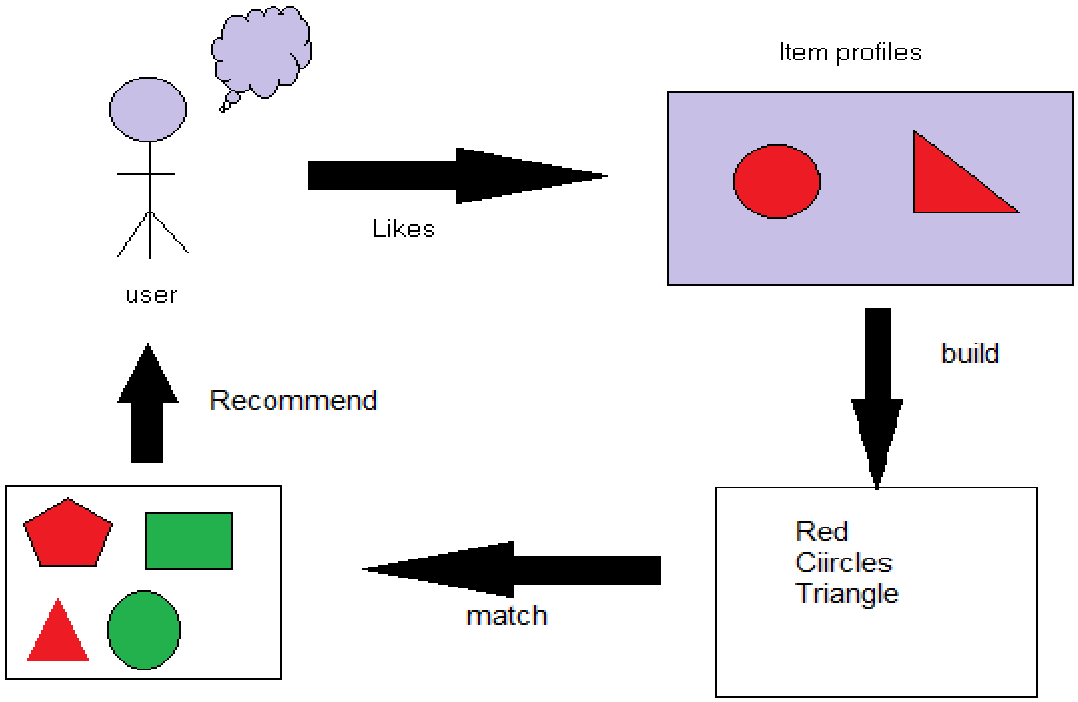

Figure 1.2

The basic workflow of content based recommendation is shown above in figure 1.2. It collects the information about the user such as what kind of things user generally likes etc. based on that it make a user profile of that information and that user profile is unique for each user in content based recommendation. Based on that user profile the algorithm or the model we used in our software predicts the user taste or behavior i.e what he/she likes? What he/she doesn't like? Etc. After successful prediction we test our model by variety of scoring methods in order to provide better recommendations to the user.

Hence the main central idea of content based recommendation is to recommend things or something else to the user like the past things assessed significantly by that user whether explicitly or implicitly. For example in case of movie recommendation system if a user watches several movies than our model should work on genres, director, actor, actress etc. in order to create a best user profile and based on that profile we generate best recommendations to our user.

1.1.3 COLLABORATIVE RECOMMENDATION SYSTEM

A collaborative type recommendation system works on the multiple user profiles. Unlike content base recommendation system it doesn't work on a single user profile. A collaborative recommendation system generally find out the similar user by using some sort of algorithms methodologies etc. and provide recommendations on the basis of that.

Like content based recommendation system it collects data implicitly and explicitly as well but the difference is that it computes that data on multiple users rather than a single user. Based on that it provides recommendations to the user.

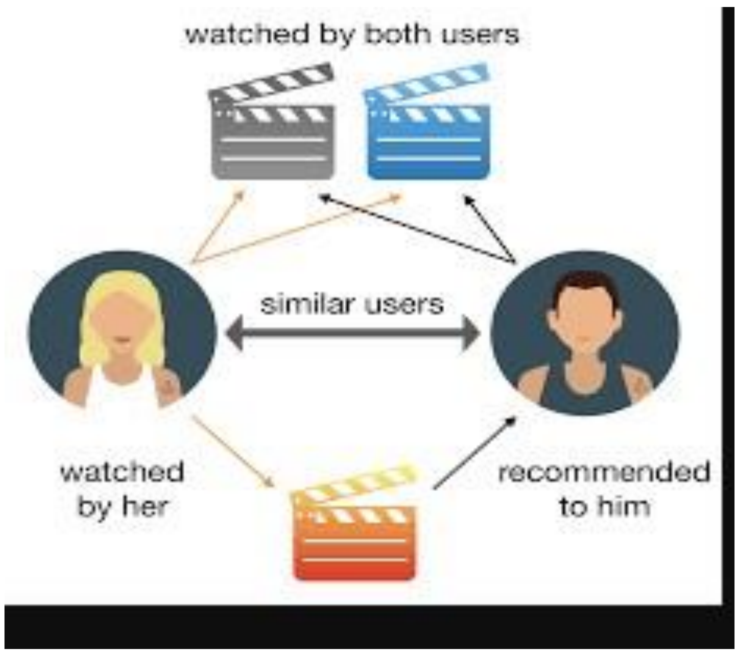

FIGURE 1.3

For example if a user watches a movie and provide it some rating than based on that rating our model find out some similar users and based on that we provide recommendation to the user as shown in above figure 1.3.We can find similar users by using different similarity measure algorithms or techniques such as

- Nearest neighbor technique
- Cosine similarity etc.

Workflow of a collaborative based recommendation:

- A customer for the most part shows his/her priority by offering stars(rating) to an item (books, motion pictures and so on.) of the application or a site.

- After collecting that data can be considered as an expected representation of the customer's enthusiasm for the comparing space.

- Based on that data our trained model find out the similar users and after finding similar users it provides recommendations to the user.

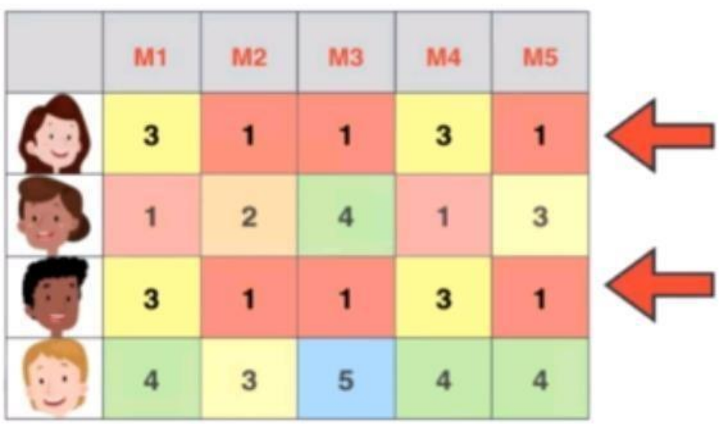

FIGURE 1.4

As from above figure we can see that user 1 and user 3 has same kind of interest(based on ratings given by them to movies). Than based on that our model thinks user1 and user3 as same user and gives same recommendations to both users.

The above technique is known as matrix factorization technique which is firstly used by Netflix in 2006, various algorithms such as nearest neighbor, dot product etc. use matrix factorization as a method. This technique handles a large amount of data. According to a survey in 2006 netflix uses a training data set of 100,480,507 ratings that 480,189 gave to 17,770 movies. Hence we can say that it works fine on large amount of dataset.

1.2 PROBLEM STATEMENT

In todays world lot of data is generated through various e commerce sites and sometimes user may be confused or waste lot of time in order to search for what he/she actually looking for. In order to make a good recommendation system it is compulsory to make a system which generally deals with the real-time environment.

1.2.1 SEARCH PROBLEM

In today's world searching is a common problem in various recommendation systems. Every user wants an exact result for which he/she looking for. No one wants to waste their time on searching. For recommendation systems this is one of the major problem. For example if a person search for a movie i.e avengers but if he/she doesn't know about the exact spelling of the movie than it is hard to find out that for what the user is actually looking for. Hence a good recommendation system is a system which generally provide best search result to the users. For example ***fuzzywuzzy*** is a method in python which is good approach to solve out this problem. Using it if someone type avengers as avngrs than it is able to auto correct it and provide results to the user on the basis of movie i.e. avenger.

1.2.2 SUGGESTION PROBLEMS

Suggestion problem is also an one of the major problem in recommendation systems. In collaborative type recommendation systems. The main causes of this problems are:

- Data sparsitry or paucity(sparse data is present in dataset)

- Cold data(no data available)

The above two problems are the main problems while making suggestions. The suggestion problem within a system cannot be solved properly but the only method to overcome from it ,is regularly improve the system. Hence we can’t fix suggestion we only improve the accuracy of our software time to time.

1.2.3 RATING PREDICTION

Rating prediction is also a very common problem in recommendation systems.For example if our mode finds out two similar user that is user1 and user2 , after some time user1 gives 4 star rating to a movie i.e movie1 than how we can exactly say that user2 also give same rating or likes that movie. In order to make a good prediction MF is a very good or an effective to handle such type of problems. MF can be implemented as Factorization machines. The main advantage of MF as FM is that we can easily add some more features or methods in order to make a good predictable recommendation system which helps users to find out the data for which they are generally looking for or search out.

Hence in order to make a good recommendation system one should regularly improve the system over a span of time , to ensure the quality, accuracy etc. of a recommendation system. Thus a good recommendation system is a real-time recommendation system.

Therefore, we should make a real-time recommendation system.

1.3 AIM AND OBJECTIVES

The main objective of this project is to make a Movie Recommendation System (in python) which generally provides movie recommendations on the basis of search(collaborative filtering) and on the basis user provide ratings(content based recommendation).

We develop content based filtering by using dot product concept which generally takes rating for each genre as an input from the user and then compute dot product of item-genre matrix and provide top -10 movie recommendations to the user.

On the other hand we develop a collaborative base recommendation system using Nearest Neighbour algorithm which takes a movie name as an input and provide top – 5 recommendations on the basis of that.

We develop a web UI by using flask that acts as an interface to the user.

The main features or objectives of our project are:

- Provide recommendations on the basis of content as well as collaborative based approaches.

- Give recommendation to the user with high accuracy by eliminating cold data or sparsity type problems.

- Eliminate search problem by using ***fuzzywuzzy*** method which also make our project more reliable.

Hence , we develop a recommendation system on both CB and CF techniques in python(on pycharm) which generally takes inputs from the user and recommends top 10 movies as recommendation to the user or client.

And flask helps us to make an interactive user interface to finish our project.

1.4 METHODOLOGY

In this project an attempt has to be made to develop a Movie Recommendation System using content based approach and collaborative base approach as well.

We develop our project on pycharm using python language.

For first approach that is Content based recommendation we use a dataset of movies which includes movies and their genres from movielens. First we develop a movie – genre matrix of our dataset. We take genres rating as an input from the user. We use dot product similarity measure in order to make a vector form of our input and gives us the similar result that is provide top 10 movie recommendations to the users.

For all the above work we make a flask user interface which takes input from the user and shows output to the user.

For the second approach that is Collaborative base recommendation we use machine learning algorithm that is nearest neighbor search which is also known as lazy algorithm because it trained all the values of dataset. We downloaded two datasets that are movies and rating datasets from movielens.

We also reduces the problems like sparsity in our project , we set a threshold frequency for both the datasets.

We use NN algorithm because it’s accuracy is very good as compared to other models or algorithms.

We also implement fuzzwuzzy method which helps us to improve search problems.

In our project we use nearest neighbor, dot product, matrix factorization etc. useful methods which is generally used by big e-comerce sites or systems such as amazon , flipkart, Netflix etc. in order to make our project as much as reliable to that sites.

We done all of our project on pycharm and jupyter notebook platform using python , html and flask language.

1.5 ORGANIZATION

1.5.1 Pycharm IDE

A pycharm is basically a integrated development environment , which is generally use for computer programming . A pycharm is just like as Anaconda Navigator but pycharm generally focuses on python programs and on the other hand anaconda has also some extra modules related to data science.

Pycharm has built in support for anaconda in it. Basically pycharm is developed for python lovers. Pycharm also provides a good framework to work with flask, django etc..and it consumes less cpu usage as compared to anaconda navigator.

The editor of pycharam is very friendly as it provides various advantages such as shows error side wise on message area screen , easy support to install modules just in a single click etc..

1.5.2 Jupyter Notebook

A jupyter notebook is an open source application on anaconda navigator. It works with a unique concepts f of cells which makes it more popular. As in pycharm you can simply download jupyter notebook.

A jupyter notebook provides a lot of advantages such as a you can share a jupyter notebook file very easily , also you can easily convert it into pdf or html extensions.

Hence a jupyter notebook provides a good framework and a good interface to the users to make there coding environment more interactive.

1.5.3 Python 3.8

Python interpreter is required in pycharm in order to run a python code. Python 3.8 is the latest version of python. It used in our project in order to compose our python code.

1.5.4 Flask

A flask is a micro framework which is written in python language.

Flask is generally used for making web- applications , writing blogs. As it is a micro framework therefore it doesn’t required any kind of libraries or modeules to run.

1.5.5 HTML

HTML stands for hyper text markup language which is generally used for designing or making of web pages. In our project we use html language with flask to make an interactive user interface.

Hence the main organizations used by us are Pycharm, Jupyter Notebook, Flask, Html and python 3.8

CHAPTER 2

LITERATURE SURVEY

Movie recommendation system :MOVREC done by Mnoj Kumar (Assistant professor, Department of IT BBDNITM Lucknow)

In today's world recommendation system plays very important role in our daily life. By going through this survey we fnd out that how MOVREC like recommendation system actually and how recommendation system rules in e-commerce sites.

Netflix	2/3rd of the movies watched are recommended
Google News	recommendations generate 38% more click-troughs
Amazon	35% sales from recommendations
Choicestream	28% of the people would buy more music if they found what they liked

FIGURE 2.1

As from above figure we can see that recommendation system plays very big role In big e-commerce sites such as amazon , Netflix etc.

MOVREC is also a movie recommendation system which provides recommendation to the user according to his taste or preferences. We find out that how a recommendation system actually works.

We also find out that how much recommendation systems are already made and what are the different techniques used in that systems such as Collabarative filtering , content based filtering etc. In 2001 a first recommender system is proposed in the market which provides recommendation on user past history. Later many recommendation systems are developed in market which uses several techniques and In 2007 Weng, Lin found that additional user profile may increase the qualities of recommendation systems. After that many recommendation systems for various purposes develops very fastly into the market.

As in earlier literature survey we discuss or came to know that how e-commerce sites earns a lot of profit just by providing good recommendations.

Similarity Measures used in Recommender Systems: A Study by Ajay Agarwal ,**Minakshi Chauhan** ,KIET group of institutions, Ghaziabad

As while searching for a particular thing there is a huge amount of data present over the internet so filtering of data is very important in order to make a good recommendation system. In order to make a recommendation system one should use various techniques but the two main techniques are:

- Content Base recommendation
- Collaborative filtering

In order to implement a recommendation using these techniques we can use any similarity measure in order to find similarty between users. This paper explains the different type of similarity measure with respect to their advantages and disadvantages also. The main similarity measures are:

- Pearson correlation
- Cosine similarity
- Mean squared differences

The above similarity measures are commonly used in various recommendation systems and also these are very helpful in order to deal with cold users (user who rated things very rarely) and also these similarity measures are very easy to implement nowadays. We find out the Cosine similarity measure in detail because on most datasets cosine similarity are used as a similarity measure.

$$similarity = cos(\theta) = \frac{\boldsymbol{u} \cdot \boldsymbol{v}}{||\boldsymbol{u}||\,||\boldsymbol{v}||} = \frac{\sum_{i=1}^{n} u_i v_i}{\sqrt{\sum_{i=1}^{n} u_i^2}\sqrt{\sum_{i=1}^{n} v_i^2}}$$

FIGURE 2.2

Experimental analysis of Recommendation System in e-commerce by Neha Verma,Bhavna Arora, Devanand

This paper includes the information that how e- commerce sites gains profit from recommendation systems and how they handled large amount of data. This paper includes that how data mining helps in making of recommendation systems. Here we came to know that a recommendation system is basically a data filtering system which filters out the data.In this article we came to know about thre major phases of a good recommendation system:

- Information Collection Phase
- Learning Phase
- Prediction Phase
- Here we came to know about that how e-commerce sites build up their recommendation systems. Basically it is done into two phases as shown in below figure.

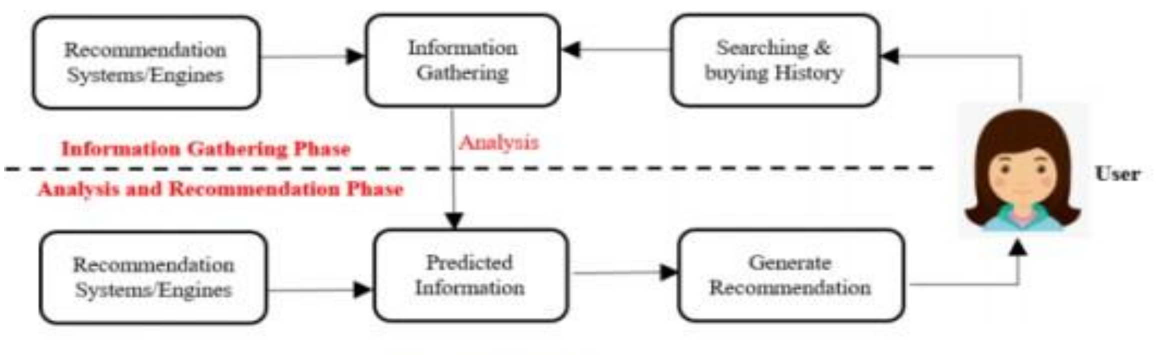

Fig.1 Workflow of RS

FIGURE 2.3

We also came to know about various techniques used by various e-commerce sites such as

- traditional techniques (clustering techniques, CB,CF etc.)
- modern techniques(peer to peer, cross domain based etc.)
- hybrid techniques(weighted method , mixed method etc.)

Hence from here we mainly focus on clustering techniques, CB techniques and on CF techniques.

Algorithms and methods in Recommender Systems by Daniar Asanov, Berlin institute of technology Berlin, Germany

In this article we find out major traditional approaches in order to make a good recommendation system. This paper describes main challenge , issues etc. while developing recommendation system. It also tells us that how data is stored atb the back end of our application and what are the different approaches to handle this data. It also gives us information regarding useful and non-useful data. It tells that how we can remove unuseful data from our dataset.

It also provide idea that how similarity measures algorithms actually works at the back end such as cosine similarity.

A cosine similarity algorithm generally represents items on a coordinate space and after that it measures the angle between the vectors and give their cosine values.

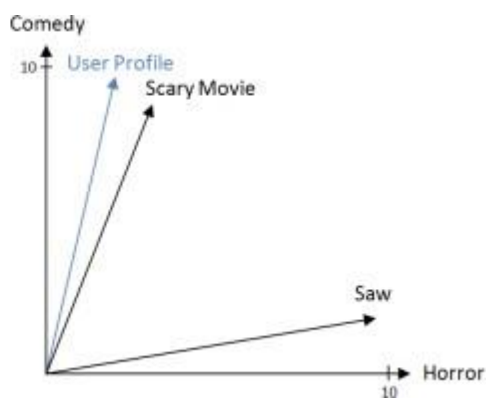

FIGURE 2.4

In it similarity is measured on the basis of angle computed by the algorithm.Lesser the angle more is the similarity. For example in above diagram user profile lies near the scary movies vector . Hence from it we can easily say that the above user generally prefer to watch scary movies as first priority and comedy movies as second priority. Also we can conclude that the above user hates to watch horror movies as horror genre make almost an orthogonal angle with above user profile (as shown in above figure). Also we study about item-based recommendation system and user-based recommendation techniques.

Issues and challenges to make a reccomender system by Shah khusro and Irfan Ullah

A recommender system basically a filtering system which generally provides filtered data or useful data to the users according to their taste , preferences etc.

As we know that there are million of users and to maintain the data of each user is a very difficult job. As we know that every human being is different is in nature and his/her choices are also different but as we know that most of the recommendation systems take care of each type of user

In this article we came to know about how this data is stored and what are the mainj issues or challenges faced by the developers in order to create a good recommender system.

The major issues are:

1. **Cold Start Problem**

   A cold start problems basically occurs when a new user add up into our database. In that case we doesn't have any idea of user taste or user preferences. In that case we can use several techniques to overcome from such type of situations. Most commonly used techniques are

   - Asking to the user to rate something at the beginning
   - Provide some questions to the user that helps in determining his taste
   - Use of collective demographic information.

2. **Sparsity**

   This problem occurs in large dataset. As while training our model for large dataset sometimes it may generate inaccurate results or recommendations. In that case we can use several techniques to overcome from such type of situations. Most commonly used techniques are:

- Demographic filtering
- SVD techniques

3. Latency Problem

Latency problem generally occours when new items added frequently and we have no ratings regarding such type of items.Using CB filtering we can reduce such type of problems but then we face another issue that is overspecialization.

Model – based collaborative filtering also helps in such type of problems

Hence from all of these literature surveys we generally know about What are recommendation Systems . How to implement a good recommendation system. What are the issues or challenges occurs while implementing the recommendation systems etc.

CHAPTER 3

SYSTEM DEVELOPMENT

3.1 REQUIREMENT ANALYSIS

The main idea or scope of our project is to develop a recommendation system . As we came to know that most of the recommendation system works on mainly two approaches that are:

- Content Based Recommendation
- Collaborative filtering

So, we develop a recoomendation system based on content as well as collaborative filtering.

Type	Definition	Example
content-based filtering	Uses *similarity between items* to recommend items similar to what the user likes.	If user A watches two cute cat videos, then the system can recommend cute animal videos to that user.
collaborative filtering	Uses *similarities between queries and items simultaneously* to provide recommendations.	If user A is similar to user B, and user B likes video 1, then the system can recommend video 1 to user A (even if user A hasn't seen any videos similar to video 1).

FIGURE 3.1

3.1.1. Content Based Recommendation:

We make a content based recommendation system which generally takes genre's rating as an input from the user and provide recommendation to the user based on that by using a similarity measure that is dot product.

As in this part of our project the main objective of our is to analyze rating given by the user to each genre with our dataset (we take dataset from the movielens) and based on that our system model computes similarity between them and provides top 10 movies as recommendation to the user.

3.1.1. Collaborative Based Recommendation:

As collaborative filtering is different from content based filtering and it is also achieve by many ways. In this project we develop a collaborative based recommendation system us using python. It takes a movie name as an input and after that it provides top 10 recomm -endations to the user. While making this we use two datasets:

- Movies dataset(downloaded from movielens)
- Rating dataset(also downloaded from movielens)

In order to make our system more reliable we sort out some major challenges that we are regularly face while making a recommender system such as: sparsity , latency problem . etc. .

By applying threshold methodologies we overcome such type of problems in our project.

Software and hardware requirements:

- Pycharm 3.5 or above
- Jupyter notebook
- Windows 7/8/10
- 64 bit operating system + 8 GB RAM
- A good web browser (like chrome , mozilla)

Hence these are some basic hardware or software requirements for our project.

3.2 DESIGN

3.2.1 CB system

We implement content based recommendation by using dot product similarity. We provideide a flask based web UI that is user interface for the user to make our projectct more interactive. The basic flow diagram of our project is given below (figure 3.2).

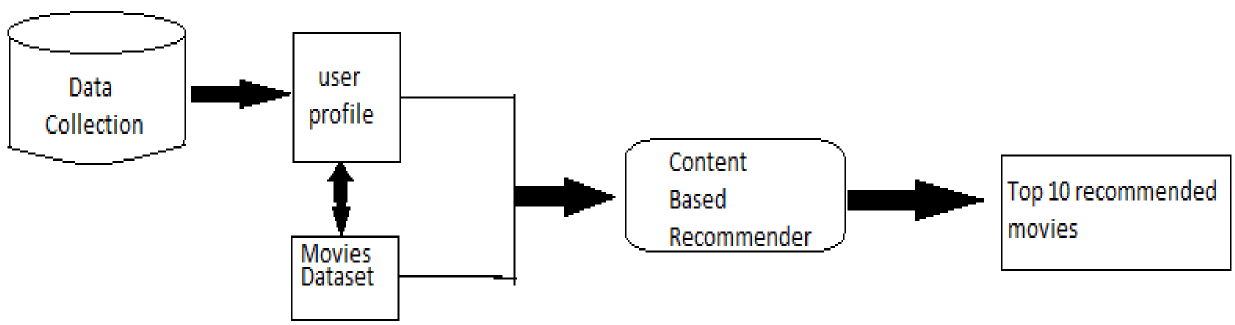

FIGURE 3.2

The above diagram shows the basic representation of our model that is content based recommendation. Firstly, we collect data or information from the user as an input i.e we collect each genre ratings from the user. After user gives rating according to his/her choice our model makes an user profile of the data that is entered by the user in the vector form.

Now, the data is in vector form form and our model already created a movie-genre matrix. Now by using dot product similarity measures our model computes the ***cosine*** angle between the vectors. At last it provides the top 10 recommended movies to the user based on his/her given preferences or taste.

3.2.1 CF system

We implement collaborative filtering recommendation system by using ***Nearest Neighbour*** algorithm. We used two datasets first one is of movies and second one is rating. We analyze both the datasets and remove all the challenges which makes our software unreliable.

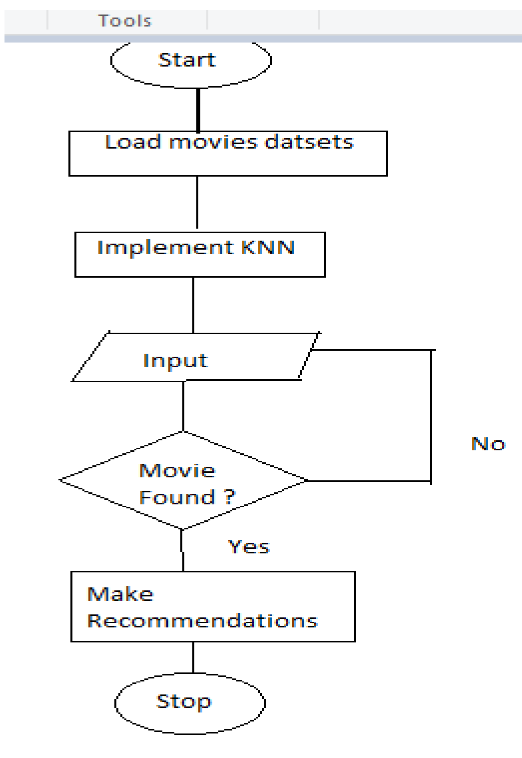

Figure 3.3

3.3 ALGORITHMS

3.2.1 Overview

CF basically provides the recommendations to the user on the basis of his preference or taste that is based on his past history or based on similar users like him and on the other hand CB provides recommendation on the basis of single user preferences or his/her taste.

In general collaborative filtering consist of m number of users , i.e. U={u1,u2,u3,u4,…..um} and a collection of n itemsi.e I={i1,i2,i3… im}. Users which uses any of items generally shows his/her behavior with respect to that particular item . These behaviors or opinions given by the user or clients can be rating given to a particular item.

Item based CF generally calculates the similarities between the item that is rated by the active user and after that it selects k most nearest neighbours and it also computes or calculates their related similarity, and provides recommendation.

In CB filtering we find out number of items U={u1,u2,u3…un} and it finds out similarity by using users past records or by using some other methods such as ask some questions to the user based on his preferences or tastes. In it preferences varies from user to user means in it we didn' wind similar user or we can say that it works on single user mechanism.

3.2.1 KNN

In order to implement or make an IB that item-based recommendation system on the basis of collaborative filtering there are many algorithms or approaches that we can follow to make a IB collaborative filtering.

But **KNN (**k- nearest neighbour**)** is a perfect model and also it is easy to implement. Many researchers says that this model i.e Knn is a very good baseline for recommender system development.

What is KNN and how it works?

KNN stands for k nearest neighbours. Basically knn is a non-parametric method and it is also known as **lazy algorithm.** Basically it utilizes a database where the information focuses are isolated into a few groups generally known as clusters to make derivation for new samples.

Knn does not make any presumptions on the hidden information dissemination however it depends on thing highlight closeness.

When knn makes deduction about a film, KNN will compute the distance between the objective film and each other film in its database ,at that point it positions its separations and returns the top k closest neighbor motion pictures or films as the most comparable film suggestions.

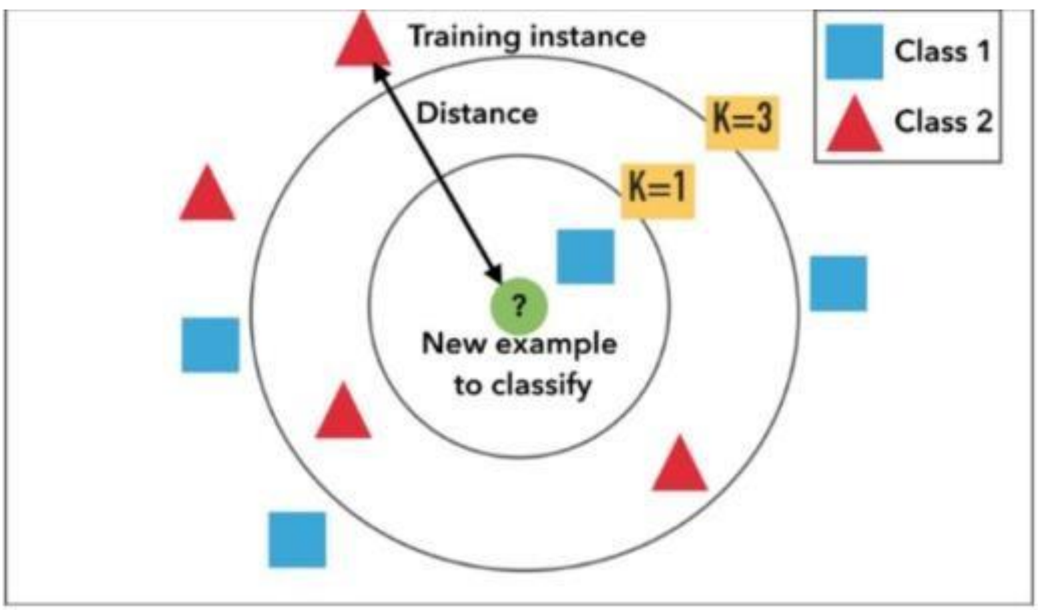

Illustration of how KNN makes classification about new sample

Figure 3.4

How to implement knn

We can use knn algorithm as two ways the first one is knn as classifier and the second one is knn as regressor. In this project we use knn as regressor we train whole of our dataset in order to obtain a maximum of accuracy in our project. But value of k also plays very important role in this. As there is no proper method to determine the value of k but in knn regressor as we train whole of our data than we can first use hit and trial method for different value of k from 1 to 10 and in our project we found at k=5 we find maximum accuracy. We use n_neighbors =5 in our project. In knn we faces multiple challenges while fitting that model into our project such as which metric we should use within it etc.

Now in order to feed the dataframe of evaluations (ratings) into this model, we need to start with dataframe of ratings, we have to change the dataframe of evaluations into an appropriate arrangements that can be devoured by a knn model.

Hence we the information to be in m*n exhibit , where m stands for the quantity of films, pictures or movies in our data and on the other hand n is the quantity of clients. In order to reshape the dataframe of evaluations , we will rotate or pivot the DF that is dataframe to the wide organization with motion pictures with pictures or movies as rows and and clients as sections or columns as shown in below diagram:

```
itemusermat=finalratings.pivot(index='movieId', columns='userId', values='rating').fillna(0)

In [19]: print(itemusermat)

userId   1    3    4    5    6    7    8    9    10   11   ...  601  602  603  \
movieId                                                    ...
1        4.0  0.0  0.0  4.0  0.0  4.5  0.0  0.0  0.0  0.0  ...  4.0  0.0  4.0
2        0.0  0.0  0.0  0.0  4.0  0.0  4.0  0.0  0.0  0.0  ...  0.0  4.0  0.0
3        4.0  0.0  0.0  0.0  5.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
5        0.0  0.0  0.0  0.0  5.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
6        4.0  0.0  0.0  0.0  4.0  0.0  0.0  0.0  0.0  5.0  ...  0.0  3.0  4.0
...      ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...
174055   0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  4.0  0.0  0.0
176371   0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  4.0  0.0  0.0
177765   0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  4.5  0.0  0.0
179819   0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
187593   0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0

userId   604  605  606  607  608  609  610
movieId
1        3.0  4.0  2.5  4.0  2.5  3.0  5.0
2        5.0  3.5  0.0  0.0  2.0  0.0  0.0
3        0.0  0.0  0.0  0.0  2.0  0.0  0.0
5        3.0  0.0  0.0  0.0  0.0  0.0  0.0
6        3.0  0.0  0.0  0.0  0.0  0.0  5.0
...      ...  ...  ...  ...  ...  ...  ...
174055   0.0  0.0  0.0  0.0  0.0  0.0  0.0
176371   0.0  0.0  0.0  0.0  0.0  0.0  0.0
177765   0.0  0.0  0.0  0.0  0.0  0.0  0.0
```

Figure 3.5

As we know that our model computes distance between the points by using eucledian distance type techniques. But eucledian distance

$$\text{Euclidean Distance} = \sqrt{(x_1 - y_1)^2 + \ldots + (x_N - y_N)^2}$$

Figure 3.6

doesn't work properly on large databases properly. As we can use knn as regressor as well as classifier , the basic difference between knn regression and in knn classification is that in knn regression it tries or generally finds out the predicted value of output variable by using a mechanism that islocal average and on the other hand knn classification tries to predict the class to which the output variable generally belongs just by calculating the local probability. We use "cosine" as metric while fitting the knn model in our project. A cosine similarity works on the vector concept.

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

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.

Figure 3.7

Basically a cosine function calculates the cosine angle between the vectors. If the angle between the two vectors is less than it means the two items are similar t each other and if the vectors makes 90 degree angle than it means there nothing similar between them.

For example in below diagram we can easily find that user profile makes smaller angle with comedy and scary movie genre it means user generally likes comedy and scary movies , so a good recommender system recommends such type of movies to the user

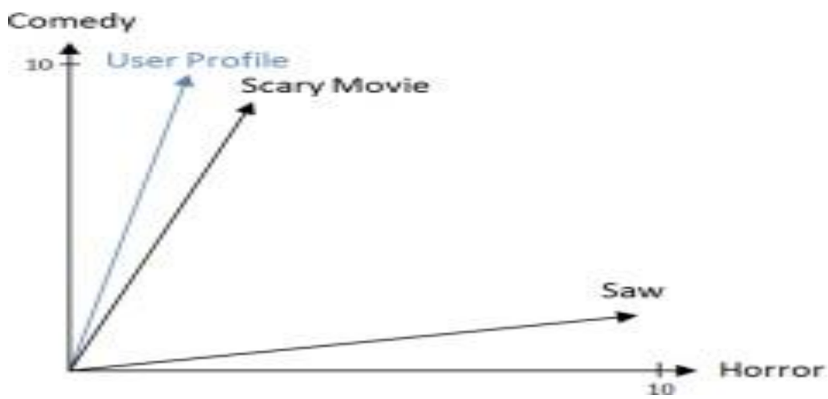

Figure 3.8

but on the other hand we can clearly see that user profile makes almost ninety degree angle with horror movies that means this user hates horror genre of movies. The main advantage of cosine similarity is that it overcomes the problems that are not handled by the eucledian measures like algorithm such as it easily deals with high - dimensional data.

3.4 BUILDING OR DEVELOPING THE PROJECT

We firstly develop a content base recommendation system and later we develop a collaborative base recommendation system.

3.4.1 CB development

We firstly use a dataset from movielens which contains movies list along with their genres. We use dot product similarity measure in this model to find out the actual taste or preferences of the user. We develop a flask based web interface to get the input from the user. Our interface basically consists of three pages:

- Welcome page

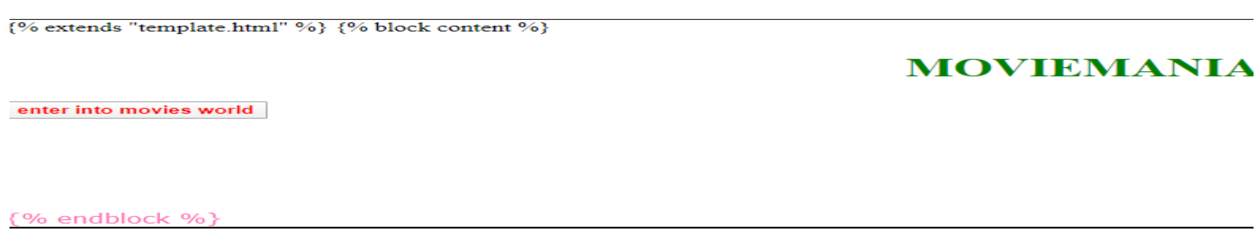

Figure 3.9

➢ Rating Page

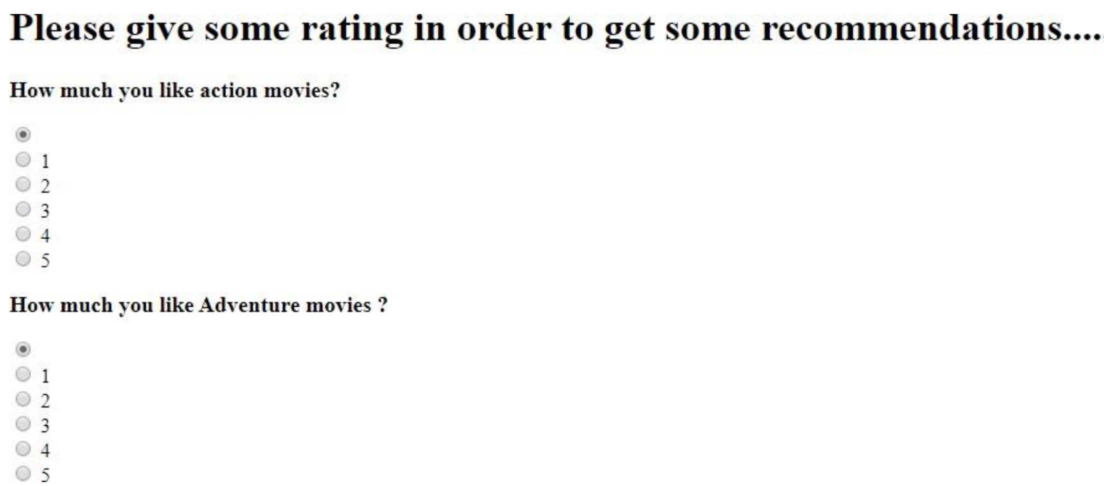

Figure 3.10

➢ Recommendation page

Figure 3.11

We create all the three web pages using HTML and connects with our python script with the help of a micro web frame-work that is flask. As full part of coding is explained in appendix section.

3.4.2 CF development

In collaborative we takes two datasets from movie lens that are :

- Movies datset
- Rating dataset

```
# reading of data sets (movies +rating)
movies=pd.read_csv('E:\\final year project data\\2nd\\datasets\\movies.csv', usecols=['movieId', 'title'],
                   dtype={'movieId': 'int32', 'title': 'str'})
ratings=pd.read_csv('E:\\final year project data\\2nd\\datasets\\ratings.csv', usecols=['userId', 'movieId', 'rating'],
                    dtype={'userId': 'int32', 'movieId': 'int32', 'rating': 'float32'})
```

Figure 3.12

As we know that while developing collaborative filtering we have multiple set of challenges such as data sparsity etc. We should take care of such type of challenges while developing our model or software.

In this project we uses plot techniques to deal with such problems as in below figure we can clearly see that number of users are very large those provide zero or no rating to the movies.

```
t[8]: Text(0, 0.5, 'count of each rating')
```

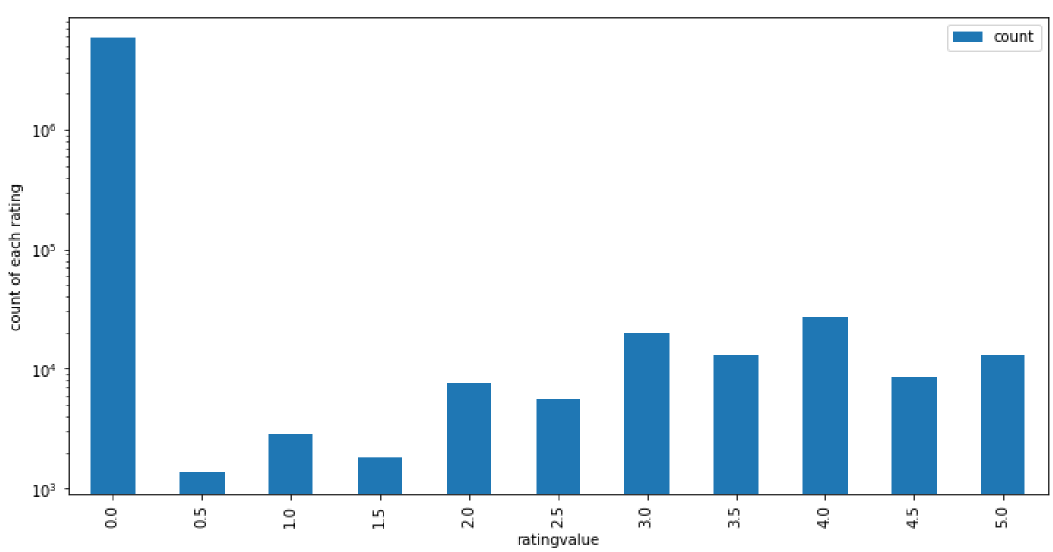

Figure 3.12

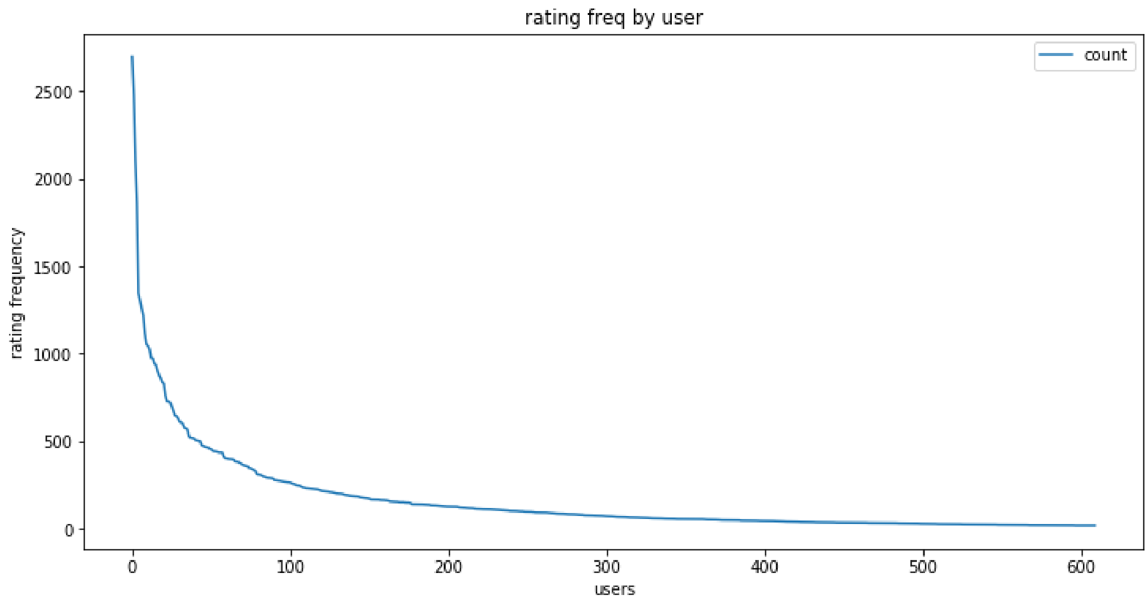

Figure 3.13

Now for above problem we use threshold concept in order to make our project more reliable. We can set a particular threshold value which deals with such type of problems in our project for this case we use threshold value =10.

After this we can clearly see the difference in shape of our dataset in figure 3.14.

```
shape of ratings with popular movies:
(81116, 3)
shape of ratings with popular movies with active user:
(78712, 3)
```

Figure 3.14

After that we have facing one of the major issue that is matrix sparsity in our project. A matrix sparsity is nothing but an unuseful data which is generally present in our dataframe. Such type of data increases the dimensions of our data frame and many of methods such eucledian distance etc. doesn't work well under this.

Hence, it leads to the curse of dimensionality. As matrix sparsity is shown in below figure that is figure 3.15.

```
In [17]: print(itemusermat)

userId   1    3    4    5    6    7    8    9    10   11   ...  601  602  603  \
movieId                                                     ...
1        4.0  0.0  0.0  4.0  0.0  4.5  0.0  0.0  0.0  0.0  ...  4.0  0.0  4.0
2        0.0  0.0  0.0  0.0  4.0  0.0  4.0  0.0  0.0  0.0  ...  0.0  4.0  0.0
3        4.0  0.0  0.0  0.0  5.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
5        0.0  0.0  0.0  0.0  5.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
6        4.0  0.0  0.0  0.0  4.0  0.0  0.0  0.0  0.0  5.0  ...  0.0  3.0  4.0
...      ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...
174055   0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  4.0  0.0  0.0
176371   0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  4.0  0.0  0.0
177765   0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  4.5  0.0  0.0
179819   0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0
187593   0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0

userId   604  605  606  607  608  609  610
movieId
1        3.0  4.0  2.5  4.0  2.5  3.0  5.0
```

Figure 3.15

As we can clearly see many zeros or non rated data in our data frame which is not useful for our algorithm so we extract only useful rows and columns by eliminating the unuseful rows and columns.

After eliminating we can see the result in fgure 3.16

```
: print(itemusermatsparse)

  (0, 0)        4.0
  (0, 3)        4.0
  (0, 5)        4.5
  (0, 13)       2.5
  (0, 15)       4.5
  (0, 16)       3.5
  (0, 17)       4.0
  (0, 19)       3.5
  (0, 23)       3.0
  (0, 27)       5.0
  (0, 28)       3.0
  (0, 29)       3.0
  (0, 34)       5.0
  (0, 37)       5.0
  (0, 38)       3.0
```

Figure 3.16

In CF we use knn algorithm (as discussed earlier) , and we we use metric as cosine because cosine similarity is very useful or you can say best fitted in big dimensional dataframes.

Fitting of knn model is shown below in figure 3.17

```
In [45]: # use of nearest neighbour algorithm
         from sklearn.neighbors import NearestNeighbors, KNeighborsClassifier
         recommendation_model = NearestNeighbors(metric='cosine', algorithm='brute',n_neighbors=5)
```

Figure 3.17

As we take movie name as an input , soothe main challenge comes here is search problem. For example if a user likes a movie that is fast and furious and user wants recommendation on the basis of fast and furious than he/ she would type fast and furious in text field and our system shows recommendation. But what happens when user type fst and furis instead of fast and furious? Then how our system works or how our system came to know that the user generally talks about fast and furious.

So, to handle such type of search problems we use ***fuzzywuzzy*** fuction in order to make our model more reliable. Implementation or definition of ***fuzzywuzzy*** function is shown below in figure 3.18.

Steps to install or add ***fuzzywuzzy*** in your list of modules:

(make surte you have connected to an internet connection)

- Type **pip install fuzzywuzzy** in acell of JN.

- In order to import type **from fuzzywuzzy import fuzz .**

- After that provide a proper definition to your function.

Hence these are the some steps to add ***fuzzywuzzy*** module into your system or environment.

```python
# use of fuzzywuzzy to make our software more reliable
from fuzzywuzzy import fuzz


def fuzzy_movie_name_matching(input_str, mapper, print_matches):
    match_movie = []
    for movie, ind in mapper.items():
        current_ratio = fuzz.ratio(movie.lower(), input_str.lower())
        if (current_ratio >= 50):
            match_movie.append((movie, ind, current_ratio))
    # sort the match_movie with respect to ratio
    # match_movie = sorted(match_movie,key =lambda x:x[2])[::-1]
    if len(match_movie) == 0:
        print("sorry......no movie found..\n")
        return -1
    if print_matches == True:
        print("some matching of input string are\n")
        for title, ind, ratio in match_movie:
            print(title, ind, '\n')

    return match_movie[0][1]
```

Figure 3.18

Figure 3.18 represents the definition of ***fuzzywuzzy*** module .

CHAPTER 4

PERFORMANCE ANALYSIS

Now in order to check performance of our developed software we need to check our result on various aspects.

4.1 CF results at various inputs

Exact input:

```
#take input from user
x = input("enter name of the movie u like \n")
```

```
enter name of the movie u like
fast and furious
```

Figure 4.1

As from figure we can clearly see that user like the movie “fast and furious” and on the basis of that our system or software provides recommendation to the user or client.

Figure 4.2 shows the output regarding this.

```
[62]: #show output
output=make_recommendation(x,itemusermatsparse,recommendation_model, movie_to_index, 5)
print(output)
```

```
on progress....

as you like fast and furious i think you also like>>>
Gone in 60 Seconds (2000)
2 Fast 2 Furious (Fast and the Furious 2, The) (2003)
Training Day (2001)
Lara Croft: Tomb Raider (2001)
Final Fantasy: The Spirits Within (2001)
```

Figure 4.2

In this case user got top five movies as recommendation (shown in figure 4.2) .

Check search problem:

Now in second we give input as wrong spelling of movie name as an input and compares the result with previous output of figure 4.2.

Case 1 :

We write fst and furis instead of fast and furious as shown in below picture or diagram:

```
In [63]: #take input from user
         x = input("enter name of the movie u like \n")

         enter name of the movie u like
         fst and furis
```

Figure 4.3

Now we compare result of this input with the previous one in order to check whether our system is flexible for such type of situations or not. Figure 4.4 and 4.5 shows the comparison between two outputs .

```
In [65]: #take input from user
         x = input("enter name of the movie u like \n")

         enter name of the movie u like
         fast and furious

In [66]: #show output
         output=make_recommendation(x,itemusermatsparse,recommendation_
         print(output)

         on progress....

         as you like fast and furious i think you also like>>>
         Gone in 60 Seconds (2000)
         2 Fast 2 Furious (Fast and the Furious 2, The) (2003)
         Training Day (2001)
         Lara Croft: Tomb Raider (2001)
         Final Fantasy: The Spirits Within (2001)
         None
```

Figure 4.4

```
In [67]: #take input from user
         x = input("enter name of the movie u like \n")

         enter name of the movie u like
         fst and furis

In [68]: #show output
         output=make_recommendation(x,itemusermatsparse,recommen
         print(output)

         on progress....

         as you like fst and furis i think you also like>>>
         Gone in 60 Seconds (2000)
         2 Fast 2 Furious (Fast and the Furious 2, The) (2003)
         Training Day (2001)
         Lara Croft: Tomb Raider (2001)
         Final Fantasy: The Spirits Within (2001)
         None
```

Figure 4.5

Hence from figure 4.4 and 4.5 we clearly see there is no difference in outputs of our model or software.

Case 2

Now in order to reconfirm lets comare the result with another one.

Now instead of writing evil dead we write evl ded and compare both the outputs.

Figure 4.6 and 4.7 shows the output of both cases respectively.

```
In [77]: #take input from user
         x = input("enter name of the movie u like \n")

         enter name of the movie u like
         evl ded

In [78]: #show output
         output=make_recommendation(x,itemusermatsparse,reco
         print(output)

         on progress....

         as you like evl ded i think you also like>>>
         Dawn of the Dead (1978)
         Bubba Ho-tep (2002)
         Evil Dead II (Dead by Dawn) (1987)
         Night of the Living Dead (1968)
         Dawn of the Dead (2004)
```

Figure 4.6(showing results for evl ded)

```
9]: #take input from user
    x = input("enter name of the movie u like \n")

    enter name of the movie u like
    evil dead

0]: #show output
    output=make_recommendation(x,itemusermatsparse,recommendation_
    print(output)

    on progress....

    as you like evil dead i think you also like>>>
    Dawn of the Dead (1978)
    Bubba Ho-tep (2002)
    Evil Dead II (Dead by Dawn) (1987)
    Night of the Living Dead (1968)
    Dawn of the Dead (2004)
```

Fogure 4.7 (showing results for evil dead)

Hence on comparing the outputs of case 1 and case 2 we can clearly see that our software shows greater flexibility and doesn’t shows any kind of search related prolems at any stage.

Check for a wrong input: (movie doesn't exist)

Now it's time to check the results for wrong input.

Figure 4.8,4.9 shows the results regarding this.

```
#take input from user
x = input("enter name of the movie u like \n")
```

```
enter name of the movie u like
fhtftdrg
```

```
#show output
output=make_recommendation(x,itemusermatsparse,
print(output)
```

```
on progress....

sorry......no movie found..
 please enter a valid movie name

pls enter a valid movie name
```

Figure 4.8

```
: #take input from user
  x = input("enter name of the movie u like \n")
```

```
enter name of the movie u like
vf gdcr gf
```

```
: #show output
  output=make_recommendation(x,itemusermatsparse,
  print(output)
```

```
on progress....

sorry......no movie found..
 please enter a valid movie name

pls enter a valid movie name
```

Figure 4.9

Hence above figure (4.8 and 4.9) shows the output results regarding a movie which doesn't exist.

4.2 CB results at various stages:

Case 1

In case 1 we only compare our results on a single genre that is user only likes horror movies.

Figure 4.10 and 4.11 shows the result of case1.

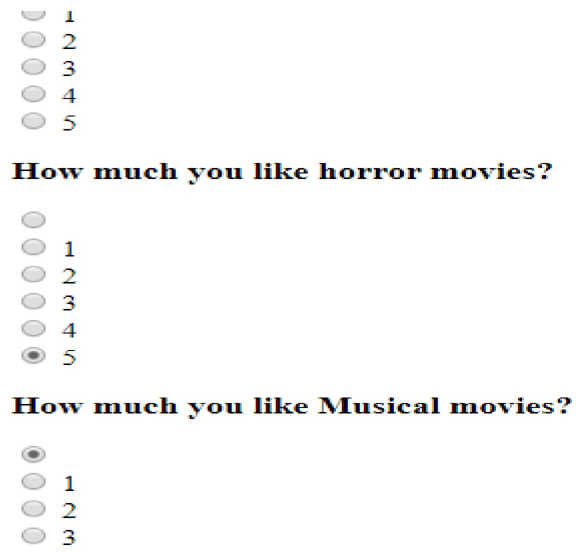

Figure 4.10(user only likes horror genre)

Result:

Fig 4.11(showing only horror movies)

Case 2

Now in this case recommends movies on multiple genres.

For example a user likes horror and comedy movies as well in that case our systemresult is:

Figure 4.12 (showing results for horror and comedy genre)

As you can clearly see that our recommendation system shows horror + comedy genres movies as a result for example as you see a movie Saturday The 14th it is one of the famous comedy horror movie.

Hence based on all above cases we analyze our model performances on each area for which we generally developed our software or project.

CHAPTER 5

CONCLUSIONS

In today’s world information increases very quickly over the internet. A large amount data is generated or added by various sites or applications everyday. There are lot of data/web pages accessible on the web in order to search or find something. And we find some data are factual and some are false. So user generally wastes lot of time in order to find the thing or item for which he/she actually looking for. A good recommender system solves such types of queries easily.

But to make a recommender system which fulfils all the customer or user expectations and provides best result is a mammoth task. Hence there is a requirement for an information base, which would assist clients with locating data and help them to choose what data to search for. This has brought about the Recommender System .

A good recommender system get familiar with the different things that clients are inspired by. It basically generates user profiles at the back end for every user , which contains the useful information about the user and with that information it predicts human behavior by using different approaches or algorithms.

Hence in today’s world , recommendation is an useful asset that can expand the adequacy of regular work, moderating a portion of the challenges engaged with finding new data and consistently checking existing data. But day by day more challenges come in front of recommendation systems . We can’t remove all the challenges at one time. We regulary make changes or provide updation to our software in order to make it more reliable. For example Netflix , amazn , flipkart etc. regularly make changes to their software and provides regular updates to the user.

5.1 Conclusion

The central idea or main purpose of our project is to make a recommendation system based on content and as well as on collaborative filtering techniques. In order to prove our project work we provides different performance analysis in previus chapter that is chapter 4.

As in today’s world most of the popular E –commerce sites uses recommender system in order to make their business more profitable. For example if we talk about the some major sites such as Netflix , amazon etc. they generate lot of profit just by recommendation.

Netflix	2/3rd of the movies watched are recommended
Amazon	35% sales from recommendations

Figure 5.1

As from figure 5.1 we can clearly see that recommendation plays very important role in modern world. As there are thousands of sites which provides recommendations for a particular item or for variety of items but only very few sites or applications are successful in order to provide or generate good recommendations.

Success of a good recommender system depends on the various factors such as advertisement of the application , product richness etc. but one of the main factor is the features that is provided by a recommender system. For a user a good recommender system should have an elegant UI design , provide useful recommendations according to his/her taste or preferences so that he/she got good recommendations in a minimum span of time.

As we develop a movie recommendation system by focusing on two main approaches that are content based recommendation (by taking genres rating as an input) and collaborative base recommendation.(user enters a movie as an input). As we know that content based recommendation works on a single user profile so there are less challenges in that case but on the other hand collaborative filtering generates lot of challenges such as search problem, cold start problem and so on. As in our project we cover most of the challenges but as we see that in today’s world data is generated very quickly over internet and it leads in increase challenges for e-commerce sites day by day. So regular updation is necessary in this research in order to overcome all the challenges.

Sometimes user doesn't want to share enough information on a site or over an application due to the some security reason and this is the one of major challenge for every recommender system.Netflix, amzazon etc. provides security to the user that there data is confidential.

Recommender structures utilizes customer data. (profiles, etc.) to fabricate modified recommendations. These structures try to make assemble of anyway much data as could sensibly be normal. This influence customer security(the system knows too much) bad. Systems, along these lines, requireto make specific and reasonable use of customer data and to guarantee a particular element of data security (non-presentation, etc.).

With everything taken into account, recommender systems despite everything require to respond to a great deal of difficulties or we can say challenges in daily life. Produced with respect to various research domains, they take huge measure of structures and ascend over various requests.

Hence we can say that this research area that is recommendation system requires to remain as wide as possible in order to recognize the most reasonable frameworks and techniques for each specific or particular application.

5.2 Future Scope

5.2.1 Incremental improvements

As we earlier discuss that no one can make a good recommendation system at one time. As it requires regular updation and regular changes **.**

As often happens, some ideas and improvements still remain to be completed. In the following sections possible future work on the recommendation system are presented.

As some of the researchers or developers find cosine similarity doesn't work well for some cases such as if we add new movie in our dataset then we didn't have enough rating for that movie at starting such type of problems can't handle by cosine ssimilarity.

In order to make an improvement many researchers works on adjusted cosine similarity. An adjusted cosine similarity is an advanced version of cosine similarity. We work on MSE in case of cosine similarity but adjusted cosine similarity works on MAE.The region of recommender frameworks is a tremendous, rising territory in the universe of online business. There is a requirement for growing more procedures for improving the precision and nature of recommender frameworks.

We can work on other preferences of user in order to improve our project because customer preferences varies every time. For example in our project we generally focuses on the movies which user likes or genres for which user want recommendation but sometime user preference is not according to the movie or a particular genre it may be different such as user only want to watch movie of a particular actor , director etc.

. So we should provide such type of incremental improvements in our project.

Hence these are the some incremental improvements which we further add in our project.

REFERENCES

1. Manoj Kumar (Assistant professor, Department of IT BBDNITM Lucknow) , “Movie Recommendation System: MOVIEREC” ,2001.

2. Ajay Agarwal ,Minakshi Chauhan (KIET group of institutions, Ghaziabad) , “Similarity Measures Used in Recommendation Systems”.

3. Neha Verma,Bhavna Arora, Devanand , “Experimental analysis of Recommendation System in e-Commerce” .

4. Daniar Asanov (Berlin Institute of Technology Berlin, Germany) , “Algorithms and Methods in Recommender Systems”.

5. Shah Khusro and Irfan Ullah , “Issues and challenges to make a Recommender System”.

6. H. Chen, A. G. Ororbia II, C. L. Giles, “ExpertSeer:A Keyphrase Based Expert Recommender for Digital Libraries”, 2015.

7. Prem Melville and Vikas Sindhwani, “ Recommender Systems”, Encyclopedia of Machine Learning, 2010

8. Gediminas Adomavicius, Nikos Manouselis, YoungOk Kwon, "Multi-Criteria Recommender Systems” , 2014.

9. X.Y. Feng, H. Zhang, Y.J. Ren, P.H. Shang, Y. Zhu, "The Deep Learning–Based Recommender System “Pubmender” for Choosing a Biomedical Publication Venue: Development and Validation Study", Journal of Medical Internet Research, 2019.

10. Naren Ramakrishnan; Benjamin J. Keller; Batul J. Mirza; Ananth Y. Grama; George Karypis, “ Privacy Risks in Recommender Systems” , 2001.

G98

ORIGINALITY REPORT

6%	1%	1%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Jaypee University of Information Technology Student Paper	2%
2	Submitted to University of Ulster Student Paper	1%
3	Submitted to University of Abertay Dundee Student Paper	<1%
4	www.ijitee.org Internet Source	<1%
5	Submitted to Queen Mary and Westfield College Student Paper	<1%

6	Submitted to Stockholms universitet Student Paper	<1%
7	docplayer.net Internet Source	<1%
8	Submitted to University of Edinburgh Student Paper	<1%
9	Submitted to University of Florida Student Paper	
10	Submitted to University of Bristol Student Paper	<1%
11	Submitted to University of Glamorgan Student Paper	<1%
12	Submitted to Heriot-Watt University Student Paper	<1%

13	Submitted to CSU, San Jose State University Student Paper	<1%
14	Bobadilla, J.. "Improving collaborative filtering recommender system results and performance using genetic algorithms", Knowledge-Based Systems, 201112 Publication	<1%
15	Oly Mistry. "Probabilistic Approaches to Tag Recommendation in a Social Bookmarking Network", Lecture Notes in Computer Science, 2012 Publication	<1%
16	Submitted to University of Lancaster Student Paper	<1%
17	gtxhdgamer.net Internet Source	<1%

Submitted to Postgraduate Institute of Medicine

Stud
ent Paper

18

---

19 Submitted to Middle East Technical University <1%

Student Paper

---

20 Chih-Ping Wei. "Web-based Recommendation Systems for Personalized e-Commerce Shopping", Integrated Series in Information Systems, 2002 <1%

Publication

---

21 Submitted to Asia Pacific University College of Technology and Innovation (UCTI) <1%

Student Paper

---

22 Submitted to University of Southampton <1%

Student Paper

---

23 Submitted to National University of Singapore <1%

Student Paper

24	Submitted to Royal Holloway and Bedford New College Student Paper	<1%
25	Submitted to Coventry University Student Paper	<1%
26	Submitted to University of Westminster Student Paper	<1%
27	Submitted to University of Wollongong Student Paper	<1%
28	Submitted to University of Greenwich Stud ent Pap er	e n t
29	Submitted to University of Northumbria at Newcastle Stud	

<1%

Pap
er

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: 15/07/2020 ✓

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | B.Tech Project Report | Paper

Name: AabhasKashyap**Department:**Computer Science **EnrolmentNo**161370

ContactNo.9459125053 **E-mail.**abhaskashyap56@gmail.com

Name oftheSupervisor: Dr. Hemraj Saini

Title of the Thesis/Dissertation/Project Report/Paper (In Capitalletters):MOVIES RECOMMENDATION SYSTEM

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages= 55
- Total No. of Preliminary pages=8
- Total No. of pages accommodate bibliography/references= 1

(Signature of Student)

FOR DEPARTMENT USE

Wehavecheckedthethesis/reportaspernormsandfound**SimilarityIndex**at .... 6(%). Therefore,we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(HemRaj Saini)

(SignatureofGuide/Supervisor) **Signature ofHOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	**Excluded**	**Similarity Index (%)**	**Generated Plagiarism Report Details** (Title, Abstract & Chapters)	
	• AllPreliminary Pages • Bibliography/Images/Quotes • 14 WordsString		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name&Signature **Librarian**

....

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: 15/07/2020 ✓

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | B.Tech Project Report | Paper

Name: Manish Sharma **Department:**Computer Science **EnrolmentNo**161377

ContactNo.7018865060 **E-mail.**manishsharmahmrhp1999@gmail.com

Name oftheSupervisor: Dr. Hemraj Saini

Title of the Thesis/Dissertation/Project Report/Paper (In Capitalletters):MOVIES RECOMMENDATION SYSTEM

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages= 55
- Total No. of Preliminary pages=8
- Total No. of pages accommodate bibliography/references= 1

(Signature of Student)

FOR DEPARTMENT USE

Wehavecheckedthethesis/reportaspernormsandfound**SimilarityIndex**at................................. 6(%). Therefore,we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(HemRaj Saini)

(SignatureofGuide/Supervisor) **Signature ofHOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
	• AllPreliminary Pages • Bibliography/Images/Quotes • 14 WordsString		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name&Signature **Librarian**

....

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com