

PERSONALIZED RECOMMENDATION SYSTEM BASED ON COLLABORATIVE FILTERING

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering / Technology degree in Computer Science

By

1) Dasari Naga Bhanu Sri(38110345)

2) Elluri Meghana (38110318)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - - 600 119

MAY – 2022



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Dasari Naga Bhanu Sri (38110345)** , **Elluri Meghana(38110318)** who carried out the project entitled **“PERSONALIZED RECOMMENDATION SYSTEM BASED ON COLLABORATIVE FILTERING”** under my supervision from December 2021 to May 2022.

Internal Guide

Dr. A. Jesudoss, M.E., Ph.D.

Head of the Department

Submitted for Viva voce Examination held on

Internal Examiner

External Examiner

(ii)

DECLARATION

I Dasari Naga Bhanu Sri hereby declare that the Project Report entitled Personalized Recommendation System Using Collaborative Filtering done by me under the guidance of Dr. . A . Jesudoss ,M.E.,Ph.d., and is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering / Technology degree in Computer Science.

DATE: PLACE:

13-03-2022
Chennai

SIGNATURE OF THE CANDIDATE

D.Naga Bhanu Sri

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of SATHYABAMA for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D, Dean**, School of Computing, **Dr. S. Vigneshwari, M.E., Ph.D.** and **Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. A. Jesudoss, M.E., Ph.D.,** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Computer Science and Engineering who were helpful in many ways for the completion of the project

PERSONALIZED RECOMMENDATION SYSTEM BASED ON COLLABORATIVE FILTERING

ABSTRACT

Recommendation technology is an important part of the Internet of Things (IoT) services, which can provide better service for users and help users get information anytime, anywhere. However, the traditional recommendation algorithms cannot meet user's fast and accurate recommended requirements in the IoT environment. In the face of a large-volume data, the method of finding neighborhood by comparing whole user information will result in low recommendation efficiency. In addition, the traditional recommendation system ignores the inherent connection between user's preference and time. In reality, the interest of the user varies over time. Recommendation system should provide users accurate and fast with the change of time. To address this, we propose a novel recommendation model based on time correlation coefficient and an improved K-means with cuckoo search (CSK-means), called TCCF. The clustering method can cluster similar users together for further quick and accurate recommendation. Moreover, an effective and personalized recommendation model based on preference pattern (PTCCF) is designed to improve the quality of TCCF. It can provide a higher quality recommendation by analyzing the user's behaviors. The extensive experiments are conducted on two real datasets of Movie Lens and Douban, and the precision of our model have improved about 5.2% compared with the MCoC model. Systematic experimental results have demonstrated our models TCCF and PTCCF are effective.

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No
	ABSTRACT	I
	LIST OF FIGURES	li
	LIST OF TABLES	lii
		iv
1	INTRODUCTION	9
2.	LITERATURE SURVEY	
	2.1	10
	2.2	11
	2.3	11
	2.4	12
	2.5	12
3	AIM AND SCOPE OF THE PRESENT INVESTIGATION	
	3.1 AIM AND OBJECTIVES	13
	3.2 PURPOSE	14
	3.3 scope	14
	3.4 EXISTING SYSTM	14

	3.4 PROPOSED SYSTEM	15
4	EXPERIMENTAL OR MATERIALS AND METHODS; ALGORITHMS USED	16
	4.1 COLLABORATIVE FILTERING	17
	4.2 SYSTEM REQUIREMENTS	19
	4.3 SOFTWARE ENVIRONMENT	23
	4.4 MODULES	31
	4.5 DATA FLOW DIAGRAM	32
	4.6 UML DIAGRAMS	33
	4.7 USE CASE DIAGRAM:	34
	4.8 ACTIVITY DIAGRAM FOR ISSUE BOOK	36
5	RESULT AND DISSCUSSION, PERFORMANCE ANALYSIS	
	5.1 REQUIREMENT ANALYSIS	37

	5.2 INPUT DESIGN AND OUTPUT DESIGN	39
	<i>5.3FEASIBILITY STUDY</i>	41
	5.4 SYSTEM TESTING	43
6		47
	SUMMARY AND CONCLUSION	
	<i>REFERENCES</i>	49
	<i>APPENDICES</i>	51

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
4.1	SYSTEM ARCHITECTURE	19
4.2	Data flow diagram	32
4.3	Use case diagram	33
4.4	Activity Diagram	3.4

CHAPTER 1

INTRODUCTION

A recommendation system or recommendation engine is a model used for information filtering where it tries to predict the preferences of a user and provide suggestions based on these preferences. These systems have become increasingly popular nowadays and are widely used today in areas such as movies, music, books, videos, clothing, restaurants, food, places and other utilities. These systems collect information about a user's preferences and behavior, and then use this information to improve their suggestions in the future. Movies are a part and parcel of life. There are different types of movies like some for entertainment, some for educational purposes, some are animated movies for children, and some are horror movies or action films. Movies can be easily differentiated through their genres like comedy, thriller, animation, action etc. Other way to distinguish among movies can be either by releasing year, language, director etc. Watching movies online, there are a number of movies to search in our most liked movies. Movie Recommendation Systems helps us to search our preferred movies among all of these different types of movies and hence reduce the trouble of spending a lot of time searching our favorable movies. So, it requires that the movie recommendation system should be very reliable and should provide us with the recommendation of movies which are exactly same or most matched with our preferences. A large number of companies are making use of recommendation systems to increase user interaction and enrich a user's shopping experience. Recommendation systems have several benefits, the most important being customer satisfaction and revenue. Movie Recommendation system is very powerful and important system. But, due to the problems associated with pure collaborative approach, movie recommendation systems also suffers with poor recommendation quality and scalability issues.

The main aim would be to develop a hybrid recommender system which incorporates and enhances properties of existing recommendation systems along with a new approach in order to decrease system runtime and to reveal latent user and item relations with great accuracy. Developing a popularity score which will help users judge the movie in a better way and success prediction for movies before release will provide better feedback to movie makers. To find a general way to make recommendation methods more effective in a broader range of applications. Although our experiments merely focus on one specific dataset, we desire to develop a universal model that can be applied to any other problem domain.

CHAPTER 2- LITERATURE SURVEY

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

2.1 Movie Recommendation System by K-Means Clustering AND K-Nearest Neighbor

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

concept that people are having common preference or choice. These users will influence on each other's opinions. This process optimizes the process and having lower RMSE.

2.2 A Review on Movie Recommendation System Using Machine Learning

System which gives recommendations usually filters the given data using various methodologies and suggests the relevant one to the customer benefit. In day-to-day life people usually use a powered recommendation system in many areas like movies, books, music, news, items etc. In this paper a wide range of work is reviewed in the field of a recommender system for movies where dataset source, methods used and accuracy are compared to deduce best one and future scope for improvement in this area are analyzed.

2.3 Movie Recommendation System Using Collaborative Filtering

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

1. Use-based filtering: User-based preferences are very common in the field of designing personalized systems. This approach is based on the user's likings. The process starts with users giving ratings (1-5) to some movies. These ratings can be implicit or explicit. Explicit ratings are when the user explicitly rates the item on some scale or indicates a thumbs-up/thumbs-down to the item. Often explicit ratings are hard to gather as not every user is much interested in providing feedbacks. In these scenarios, we gather implicit ratings based on their behavior. For instance, if a user buys a product more than once, it indicates a positive preference. In context to movie systems, we can imply that if a user watches the entire movie, he/she has some likeability to it. Note that there are no clear rules in determining implicit ratings. Next, for each user, we first find some defined number of nearest neighbors. We calculate

correlation between users' ratings using Pearson Correlation algorithm. The assumption that if two users' ratings are highly correlated, then these two users must enjoy similar items and products are used to recommend items to users.

2. Item-based filtering: Unlike the user-based filtering method, item based focuses on the similarity between the item's users like instead of the users themselves. The most similar items are computed ahead of time. Then for recommendation, the items that are most similar to the target item are recommended to the user.

2.4 Products and Movie Recommendation System for Social Networking Sites

Recommendation systems are an integral part of information filtering system in data science, that are widely used in order to identify the pattern a user would likely choose on the basis of the previous choices of the user as well as from studying the pattern in which others have chosen. For a fact, the recommendation can never be a cent percent correct at providing recommendations to the user but can be close enough to please them to a certain extent. Thus, the same is widely used in the industries these days to get higher profit and have a good hold in the market. The data scientists of every company design some algorithm that studies the information from the social network and clusters the data. There can be a single algorithm for classification like k-Means clustering or Hidden Markov model or can be done by bagging and boosting techniques. With this technique of displaying the movies or products into the profile of a particular customer, they not only increase their business but also enhance the customer experiences but there are several issues related to the standard techniques like the cold start problem, shrill attack, etc. thereby increasing the scope of research in this field. This work deals with both Collaborative Filtering and Content-Based Filtering to form a product and movie recommendation system for the social networking sites that shows the effectiveness of collaborative filtering and portrays the challenges faced by content-based filtering.

2.5 An Intelligent Movie Recommendation System based on user priorities

Today's web and app users request modified experiences. They anticipate the apps, news sites, social networks they engage with to evoke who they are and what they're fascinated in and make related, adjusted, and accurate commendations for new content and new goods based on their earlier deeds. This can be done using Recommended Systems in Machine Learning. In this paper we use Recommender System to recommend movies based on his previous ratings on movie he came across.

CHAPTER 3

AIM AND SCOPE OF THE PRESENT INVESTIGATION

3.1 AIM AND OBJECTIVES

The main aim would be to develop a hybrid recommender system which incorporates and enhances properties of existing recommendation systems. Developing a popularity score which will help users judge the movie in a better way .To find a general way to make recommendation methods more effective in a broader range of applications . A new approach in order to decrease system runtime and to reveal latent user and item relations with great accuracy.

Users do not always leave behind enough personalized information along their customer journey. For instance, new customers can be acquired or existing customers might browse an e-commerce website without being logged in. Non-personalized recommendation systems, such as those based on proposals for products frequently purchased together, still offer recommendation opportunities for companies in this case. However, the more individually these are tailored to the customer, the better. Therefore, in the following, personalized approaches are presented which learn the preferences of customers. To understand these methods, it is helpful to consider the recommendation problem at hand as a sparsely populated matrix. The rows represent the users, the columns represent the items. Whenever a user performs an action for an item, a

respective entry is recorded in the matrix. Otherwise, values remain absent and do not need to be explicitly stored.

3.2 PURPOSE

The purpose of this document is to develop a hybrid recommender system which incorporates and enhances properties of existing recommendation systems along with a new approach in order to decrease system runtime and to reveal latent user and item relations with great accuracy. In detail, this document will provide a general description of our project, including user requirements, product perspective, and overview of requirements, general constraints. In addition, it will also provide the specific requirements and functionality needed for this project - such as interface, functional requirements and performance requirements.

3.3 SCOPE

The scope of this SRS document persists for the entire life cycle of the project. This document defines the final state of the software requirements agreed upon by the customers and designers. Finally at the end of the project execution all the functionalities may be traceable from the SRS to the product. The document describes the functionality, performance, constraints, interface and reliability for the entire life cycle of the project.

3.4 EXISTING SYSTEM

Gharbi Alshammari and Stelios Kapetanakis (July 2019) proposed a method that proposes grading and location-based in combination of attributes with CF with a problem of scarcity. Also this method stated similarities in large data set with no pre-operations. This used 1 million data set from movie lens. They also applied methods like Random forest, KNN. As a conclusion performance is improved when Hybrid method is combined with location-based approach.

3.4 PROPOSED SYSTEM

They proposed a movie recommendation system using collaborative filtering that focuses on the ratings given by the users to provide recommendations. The proposed system is built using machine learning algorithm to sort the movies according to the ratings. In one paper the authors propose a fully content-based movie recommendation system to recommend movies. The proposed system makes use of a neural network with the content information of the movies to obtain features and learn the similarities between movies. The authors implement a recommendation system that combines both user-based and item-based collaborative filtering approach. The system is built using machine learning technique and develop a new algorithm that unifies used based and item-based recommendations. Based on the research we conducted, collaborative filtering was found to be one of the popularly used approaches to build recommendation systems.

CHAPTER 4

EXPERIMENTAL OR MATERIALS AND METHODS; ALGORITHMS USED

Recommender systems are information filtering systems that help deal with the problem of information overload by filtering and segregating information and creating fragments out of large amounts of dynamically generated information according to user's preferences, interests, or observed behavior about a particular item or items. A Recommender system has the ability to predict whether a particular user would prefer an item or not based on the user's profile and its historical information. Recommendation systems have also proved to improve the decision making processes and quality. In large e-commerce settings, recommender systems enhance the revenues for marketing, for the fact that they are effective means of selling more products. In scientific libraries, recommender systems support and allow users to move beyond the generic catalogue searches. Therefore, the need to use efficient and accurate recommendation techniques within a system that provides relevant and dependable recommendations for users cannot be neglected.

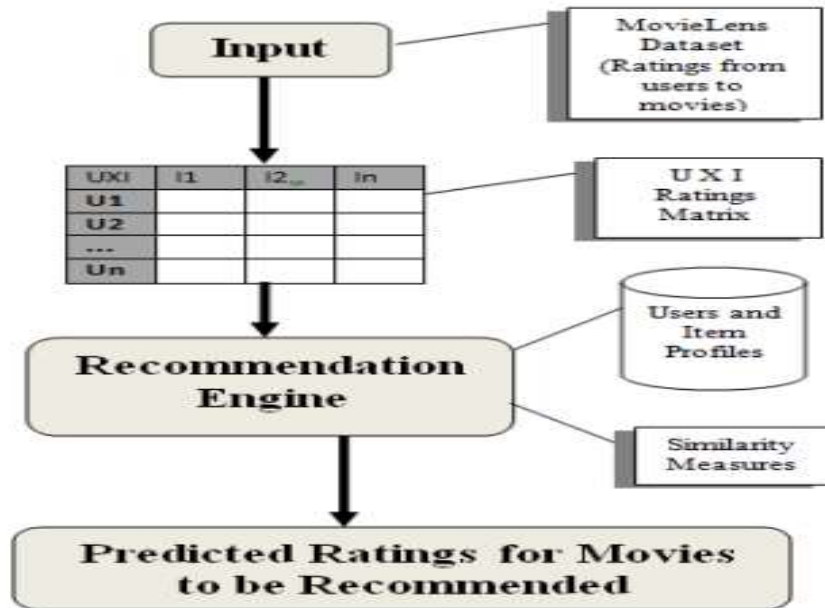
Conglomerates like Netflix use a recommendation engine to present their viewers with movie and show suggestions. Amazon, on the other hand, uses its recommendation engine to present customers with product recommendations. While each uses the one for slightly different purposes, both in general have the same goal: to drive sales, boost engagement and customer retention, and deliver more personalized customer experiences. Recommendations typically speed up the searches and make it easier for users to access the content they have always been interested in, and surprise them with several offers they would have never searched for. Doing companies are able to gain new customers by sending out customized emails with links to new offers that meet the recipient interests or suggestions of films and TV shows that suit their particular profiles.

4.1 COLLABORATIVE FILTERING

The Collaborative filtering method for recommender systems is a method that is solely based on the past interactions that have been recorded between users and items, in order to produce new recommendations. Collaborative Filtering tends to find what similar users would like and the recommendations to be provided and in order to classify the users into clusters of similar types and recommend each user according to the preference of its cluster. The main idea that governs the collaborative methods is that through past user-item interactions when processed through the system, it becomes sufficient to detect similar users or similar items to make predictions based on these estimated facts and insights.

Such memory-based approaches directly work with the values of recorded interactions or data and are essentially core based on nearest neighbors search, finding the closest users from a user of interest and suggest the most popular items among these neighbors. The created model approaches assuming there is an underlying “generative” insight that explains the user-item interactions and tries to discover it in order to make new predictions. It recommends an item to user A based on the interests of a similar user B. Furthermore, the embeddings can be learned automatically, without relying on hand-engineering of features. The collaborative filtering method does not need the features of the items to be given. Every user and item is described by a feature vector or embedding.

Fig 4.1 :- SYSTEM ARCHITECTURE



4.2 SYSTEM REQUIREMENTS

4.2.1 HARDWARE REQUIREMENTS

System: Pentium Dual Core

Hard Disk: 120 GB

Monitor: 15" LED

Input Devices: Keyboard, Mouse

Ram: 4 GB

4.2.2 SOFTWARE REQUIREMENTS

Operating system: Windows 7/10

Coding Language: Python

4.3 SOFTWARE ENVIRONMENT

4.3.1 Python:

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- Python is Interpreted – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive – You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

- Python is Object-Oriented – Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- Python is a Beginner's Language – Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

4.3.2 History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Smalltalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

4.3.4 Python Features

Python's features include –

- **Easy-to-learn** – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** – Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** – Python's source code is fairly easy-to-maintain.
- **A broad standard library** – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

- Portable – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- Extendable – You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- Databases – Python provides interfaces to all major commercial databases.
- GUI Programming – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- Scalable – Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below –

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

Getting Python

The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python <https://www.python.org>.

Windows Installation

Here are the steps to install Python on Windows machine.

- Open a Web browser and go to <https://www.python.org/downloads/>.
- Follow the link for the Windows installer python-XYZ.msifile where XYZ is the version you need to install.
- To use this installer python-XYZ.msi, the Windows system must support Microsoft Installer 2.0. Save the installer file to your local machine and then run it to find out if your machine supports MSI.
- Run the downloaded file. This brings up the Python install wizard, which is really easy to use. Just accept the default settings, wait until the install is finished, and you are done.

The Python language has many similarities to Perl, C, and Java. However, there are some definite differences between the languages.

First Python Program

Let us execute programs in different modes of programming.

Interactive Mode Programming

Invoking the interpreter without passing a script file as a parameter brings up the following prompt –

```
$ python
Python2.4.3(#1,Nov112010,13:34:43)
[GCC 4.1.220080704(RedHat4.1.2-48)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Type the following text at the Python prompt and press the Enter –

```
>>>print"Hello, Python!"
```

If you are running new version of Python, then you would need to use print statement with parenthesis as in **print ("Hello, Python!");**. However in Python version 2.4.3, this produces the following result –

```
Hello, Python!
```

Script Mode Programming

Invoking the interpreter with a script parameter begins execution of the script and continues until the script is finished. When the script is finished, the interpreter is no longer active.

Let us write a simple Python program in a script. Python files have extension **.py**. Type the following source code in a test.py file –

```
print"Hello, Python!"
```

We assume that you have Python interpreter set in PATH variable. Now, try to run this program as follows –

```
$ python test.py
```

This produces the following result –

```
Hello, Python!
```

Flask Framework:

Flask is a web application framework written in Python. Armin Ronacher, who leads an international group of Python enthusiasts named Pocco, develops it. Flask is based on Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects.

Http protocol is the foundation of data communication in world wide web. Different methods of data retrieval from specified URL are defined in this protocol.

The following table summarizes different http methods –

Sr.No	Methods & Description
1	GET Sends data in unencrypted form to the server. Most common method.
2	HEAD Same as GET, but without response body
3	POST Used to send HTML form data to server. Data received by POST method is not cached by server.
4	PUT Replaces all current representations of the target resource with the uploaded content.
5	DELETE Removes all current representations of the target resource given by a URL

By default, the Flask route responds to the **GET** requests. However, this preference can be altered by providing methods argument to **route()** decorator.

In order to demonstrate the use of **POST** method in URL routing, first let us create an HTML form and use the **POST** method to send form data to a URL.

Save the following script as login.html

```
<html>

<body>

<form action="http://localhost:5000/login" method="post">

<p>Enter Name:</p>

<p><input type="text" name="nm"/></p>

<p><input type="submit" value="submit"/></p>

</form>

</body>

</html>
```

Now enter the following script in Python shell.

```
from flask import Flask, redirect, url_for, request

app = Flask(__name__)

@app.route('/success/<name>')

def success(name):

    return 'welcome %s' % name

@app.route('/login', methods=['POST', 'GET'])

def login():

    if request.method == 'POST':

        user = request.form['nm']
```

```
return redirect(url_for('success',name= user))

else:

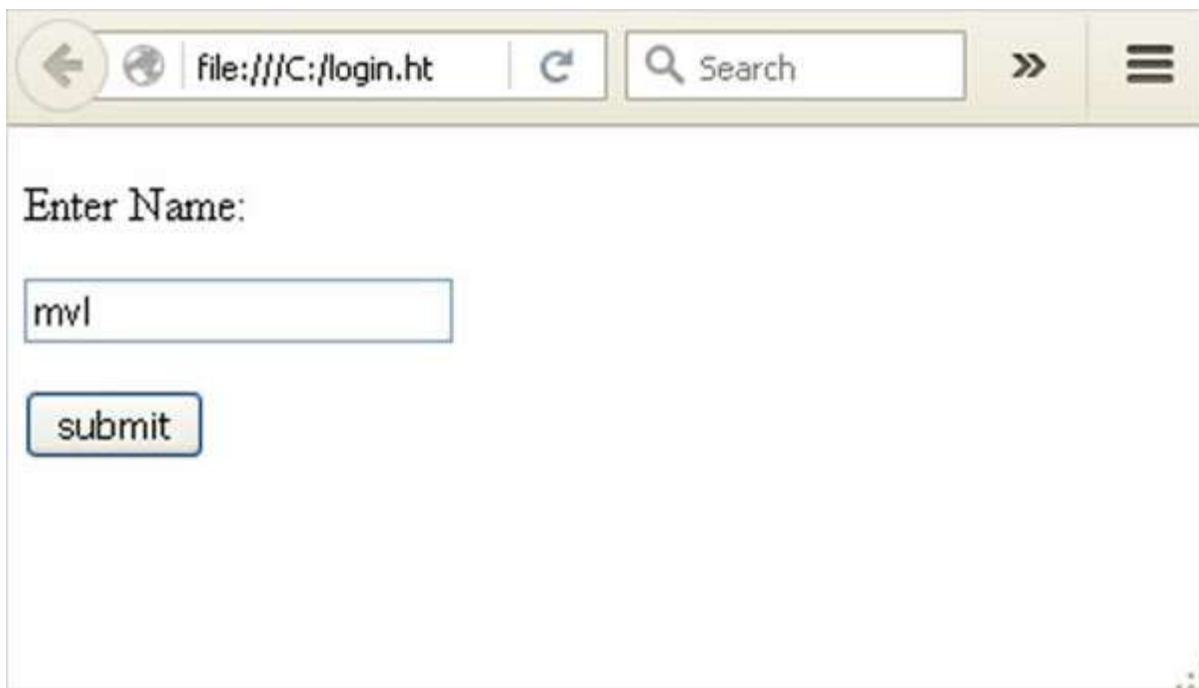
user=request.args.get('nm')

return redirect(url_for('success',name= user))

if __name__ == '__main__':

app.run(debug =True)
```

After the development server starts running, open **login.html** in the browser, enter name in the text field and click **Submit**.

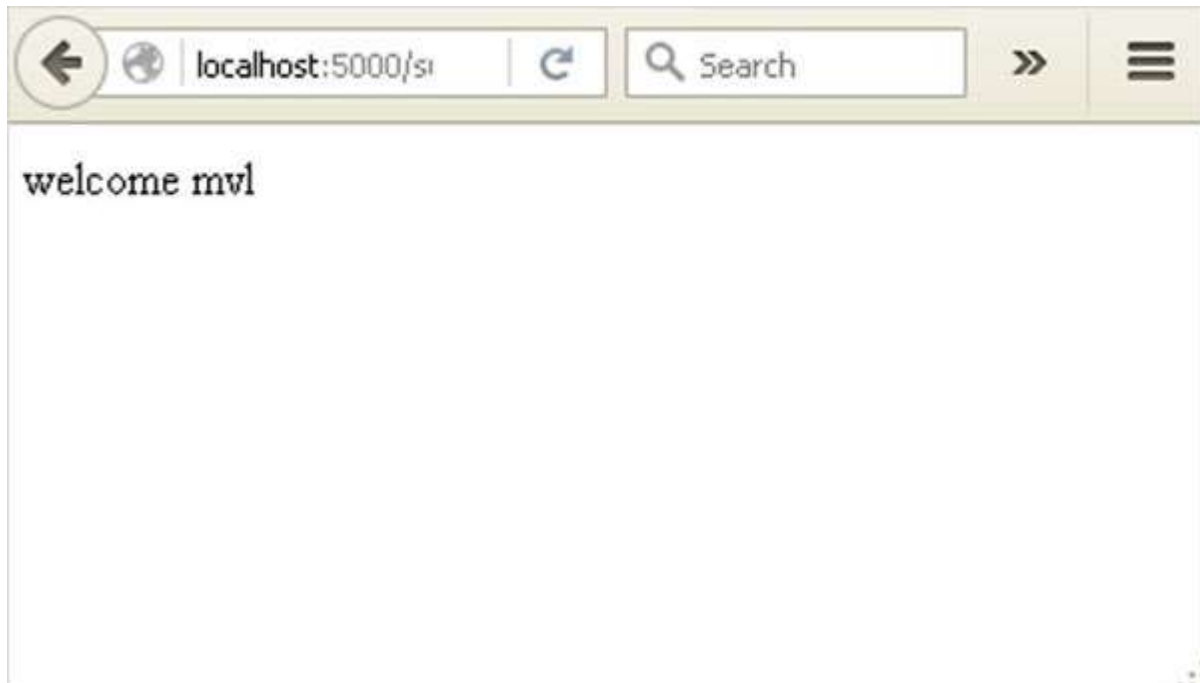
A screenshot of a web browser window. The address bar shows the file path 'file:///C:/login.ht'. The page content includes a label 'Enter Name:', a text input field with the value 'mvl', and a button labeled 'submit'.

Form data is POSTed to the URL in action clause of form tag.

http://localhost/login is mapped to the **login()** function. Since the server has received data by **POST** method, value of 'nm' parameter obtained from the form data is obtained by –

```
user = request.form['nm']
```

It is passed to **'/success'** URL as variable part. The browser displays a **welcome** message in the window.



Change the method parameter to **'GET'** in **login.html** and open it again in the browser. The data received on server is by the **GET** method. The value of 'nm' parameter is now obtained by –

```
User = request.args.get('nm')
```

Here, **args** is dictionary object containing a list of pairs of form parameter and its corresponding value. The value corresponding to 'nm' parameter is passed on to **'/success'** URL as before.

What is Python?

Python is a popular programming language. It was created in 1991 by Guido van Rossum.

It is used for:

- web development (server-side),
- software development,

- mathematics,
- System scripting.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-orientated way or a functional way.

Good to know

- The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.
- In this tutorial Python will be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.

Python Syntax compared to other programming languages

- Python was designed to for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.

- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

Python Install

Many PCs and Macs will have python already installed.

To check if you have python installed on a Windows PC, search in the start bar for Python or run the following on the Command Line (cmd.exe):

```
C:\Users\Your Name>python --version
```

To check if you have python installed on a Linux or Mac, then on linux open the command line or on Mac open the Terminal and type:

```
python --version
```

If you find that you do not have python installed on your computer, then you can download it for free from the following website: <https://www.python.org/>

Python Quickstart

Python is an interpreted programming language, this means that as a developer you write Python (.py) files in a text editor and then put those files into the python interpreter to be executed.

The way to run a python file is like this on the command line:

```
C:\Users\Your Name>python helloworld.py
```

Where "helloworld.py" is the name of your python file.

Let's write our first Python file, called helloworld.py, which can be done in any text editor.

```
helloworld.py  
  
print("Hello, World!")
```

Simple as that. Save your file. Open your command line, navigate to the directory where you saved your file, and run:

```
C:\Users\Your Name>python helloworld.py
```

The output should read:

```
Hello, World!
```

Congratulations, you have written and executed your first Python program.

The Python Command Line

To test a short amount of code in python sometimes it is quickest and easiest not to write the code in a file. This is made possible because Python can be run as a command line itself.

Type the following on the Windows, Mac or Linux command line:

```
C:\Users\Your Name>python
```

From there you can write any python, including our hello world example from earlier in the tutorial:

```
C:\Users\Your Name>python
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>print("Hello, World!")
```

Which will write "Hello, World!" in the command line:

```
C:\Users\Your Name>python
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>print("Hello, World!")
Hello, World!
```

Whenever you are done in the python command line, you can simply type the following to quit the python command line interface:

```
exit()
```

Execute Python Syntax

As we learned in the previous page, Python syntax can be executed by writing directly in the Command Line:

```
>>>print("Hello,                               World!")  
Hello, World!
```

Or by creating a python file on the server, using the .py file extension, and running it in the Command Line:

```
C:\Users\Your Name>python myfile.py
```

4.4 MODULES

- Data Collection Module
- Data Pre-Processing Module
- Recommendation Module
- Evaluation Module

4.4.1 Data Collection Module

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

4.4.2 Data Pre-Processing Module

we preprocess the datasets into a proper format and transform these data frame of ratings to a suitable format for our model. We want the data to be in an $m \times n$ matrix, where m and n gives number of movies and users.

4.4.4 Recommendation Module

In this Module, we have modeled different collaborative filters to predict ratings for users and the distance between the targeted items and other items is obtained by similarity measure, which gives us the top machine learning and finally predicting the required recommended list of movies with decreasing order of distance.

4.4.5 Evaluation Module

Evaluation is done based on the actual ratings and predicted ratings comparison of the user which were already seen and present in remaining dataset and similar process is repeated with other similarity measures.

4.5 DATA FLOW DIAGRAM

Data Flow Diagram (DFD) is a two-dimensional diagram that describes how data is processed and transmitted in a system. The graphical depiction recognizes each source of data and how it interacts with other data sources to reach a mutual output. In order to draft a data flow diagram one must

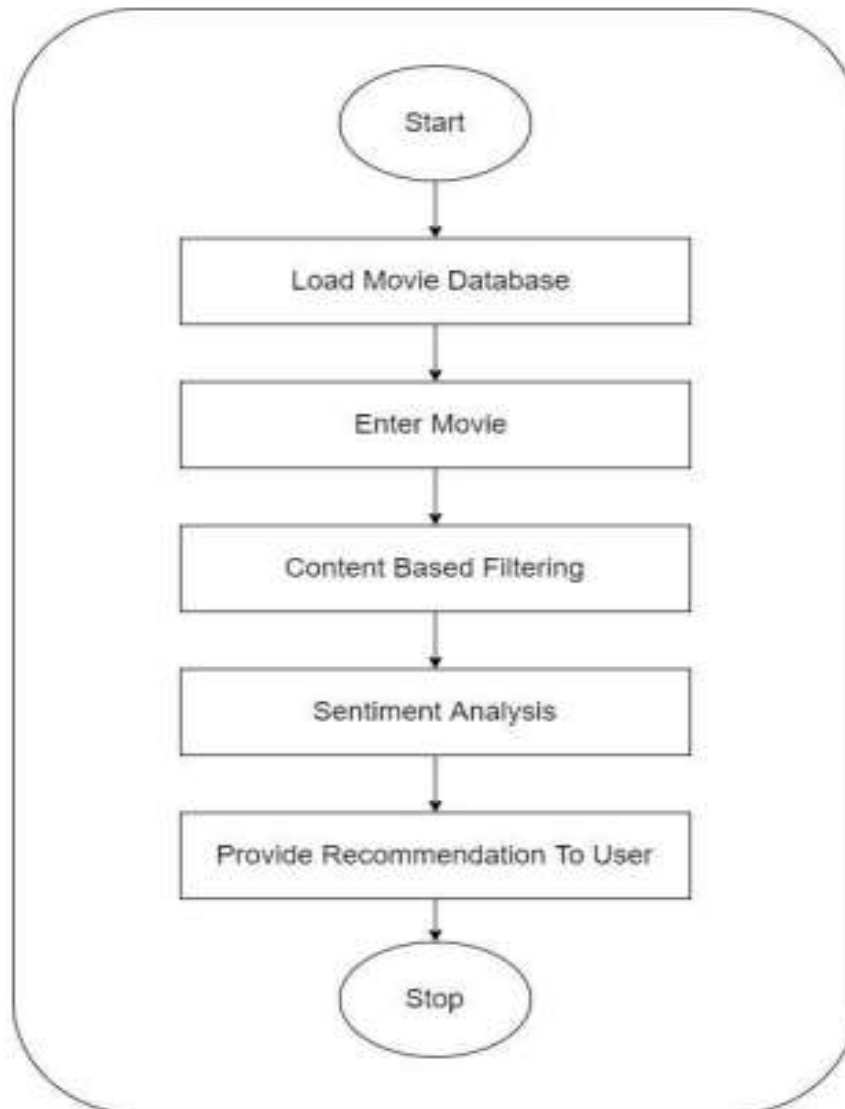
- Identify external inputs and outputs
- Determine how the inputs and outputs relate to each other
- Explain with graphics how these connections relate and what they result in.

4.5.1 Role of DFD:

- It is a documentation support which is understood by both programmers and nonprogrammers. As DFD postulates only what processes are accomplished not how they are performed.

- A physical DFD postulates where the data flows and who processes the data.
- It permits analyst to isolate areas of interest in the organization and study them by examining the data that enter the process and viewing how they are altered when they leave.

Figure 4.1



4.6 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major

components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

4.7 USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those

use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



Fig 4.2

4.8 ACTIVITY DIAGRAM FOR ISSUE BOOK:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

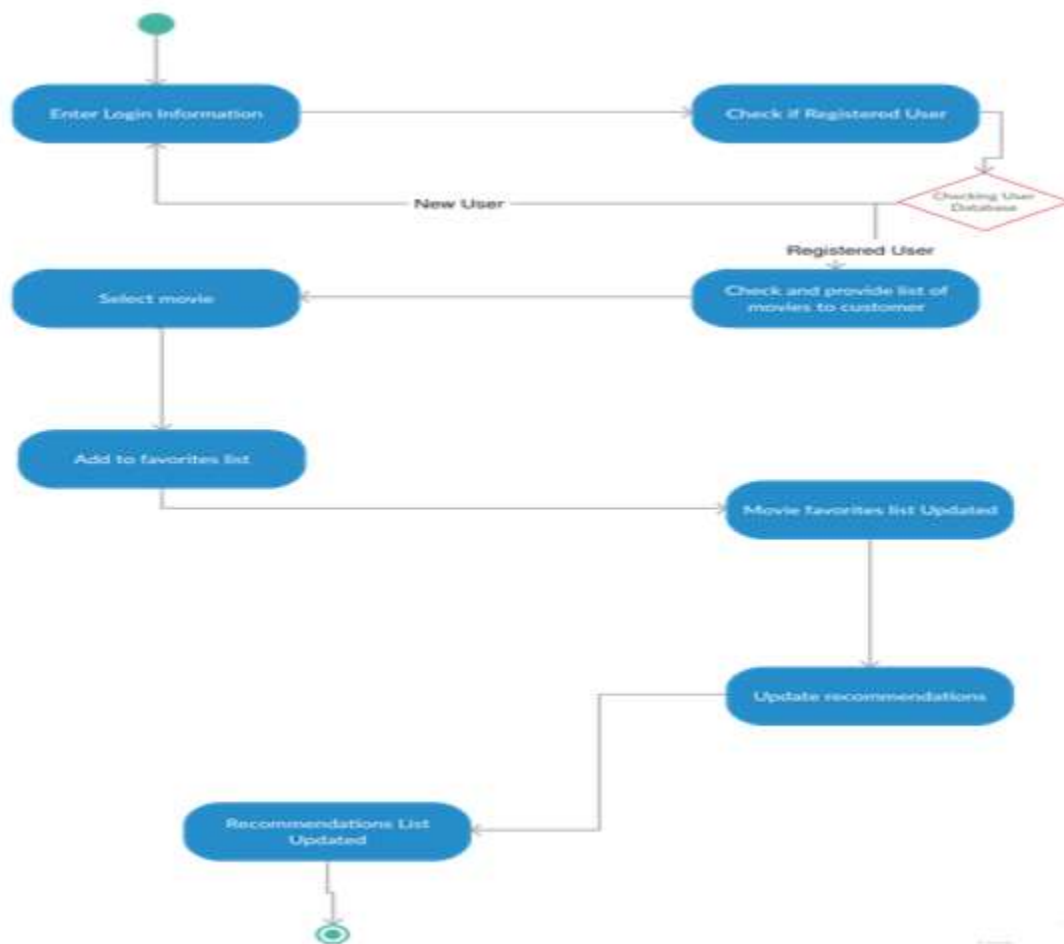


Fig 4.

CHAPTER 5

RESULT AND DISSCUSSION, PERFORMANCE ANALYSIS

5.1 REQUIREMENT ANALYSIS

Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed. In software engineering, such requirements are often called functional specifications. Requirements analysis is an important aspect of project management.

5.1.1 FUNCTIONAL REQUIREMENTS

It is a technical specification requirement for the software products. It is the first step in the requirement analysis process which lists the requirements of particular software systems including functional, performance and security requirements. The function of the system depends mainly on the quality hardware used to run the software with given functionality.

Usability

It specifies how easy the system must be use. It is easy to ask queries in any format which is short or long, porter stemming algorithm stimulates the desired response for user.

Robustness

It refers to a program that performs well not only under ordinary conditions but also under unusual conditions. It is the ability of the user to cope with errors for irrelevant queries during execution.

Security

The state of providing protected access to resource is security. The system provides good security and unauthorized users cannot access the system there by providing high security.

Reliability

It is the probability of how often the software fails. The measurement is often expressed in MTBF (Mean Time between Failures). The requirement is needed in order to ensure that the processes work correctly and completely without being aborted. It can handle any load and survive and even capable of working around any failure.

Compatibility

It is supported by version above all web browsers. Using any web servers like localhost makes the system real-time experience.

Flexibility

The flexibility of the project is provided in such a way that it has the ability to run on different environments being executed by different users.

Safety

Safety is a measure taken to prevent trouble. Every query is processed in a secured manner without letting others to know one's personal information.

5.1.2 NON- FUNCTIONAL REQUIREMENTS

Portability

It is the usability of the same software in different environments. The project can be run in any operating system.

Performance

These requirements determine the resources required, time interval, throughput and everything that deals with the performance of the system.

Accuracy

The result of the requesting query is very accurate and high speed of retrieving information. The degree of security provided by the system is high and effective.

Maintainability

Project is simple as further updates can be easily done without affecting its stability. Maintainability basically defines that how easy it is to maintain the system. It means that how easy it is to maintain the system, analyse, change and test the application. Maintainability of this project is simple as further updates can be easily done without affecting its stability.

5.2 INPUT DESIGN AND OUTPUT DESIGN

5.2.1 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

5.2.2 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- ❖ Convey information about past activities, current status or projections of the
- ❖ Future.
- ❖ Signal important events, opportunities, problems, or warnings.
- ❖ Trigger an action.
- ❖ Confirm an action.

5.3FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is

to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ◆ ECONOMICAL FEASIBILITY
- ◆ TECHNICAL FEASIBILITY
- ◆ SOCIAL FEASIBILITY

5.3.4 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

5.3.5 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

5.3.6 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

5.4 SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

5.4.1 TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of

individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

5.4.2 Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

5.4.3 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

5.4.4 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements

document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

CHAPTER 6

SUMMARY AND CONCLUSION

In this project, we have discussed that how the movie tweets have been collected from micro blogging websites to understand the current trends and user response of the movie and experiments conducted on public database produce promising results. The proposed system is also scalable for handling enormous amount of information available online. The system is not having complex process to recommend the movies than the existing system. Proposed system gives genuine and fast result than existing system.

REFERENCES

- [1] Lavanya, R., TanmayKhokle, AbhideepMaity. (2020). "Review on Hybrid Recommender System for Mobile Devices" Springer Lecture notes on networks and systems, 130, 477-486.
- [2] Liaoliang Jiang, Yuting Cheng, L. Y. J. L. H. Y. X. W. (2019). "A trust based collaborative filtering algorithm for e commerce recommendation system." Journal of Ambient Intelligence and Humanized Computing, 10, 3023–3034.
- [3] Lavanya, R., RithikaLahari, PalakGupta. (2019). "An Optimal Enhancement of the Dynamic Features of Recommender Systems" International Journal of Recent Technology and Engineering, 8(2S4), 51-55.
- [4] GharbiAlshammari, SteliosKapetanakis, Abdullah Alshammari and NikolaosPolatidis "Improved Movie Recommendations Based on a Hybrid Feature Combination Method", International Journal Of Engineering And Computer Science, July 2019.
- [5] Yi Ren, JingkeXu, Jie Huang and Cuirong Chi " Research on Collaborative Filtering Recommendation Algorithm for Personalized Recommendation System", 2019 9th International Conference on Education and Social Science (ICESS 2019).
- [6] Vishwa, Bhavesh, Aman Gupta, PranalSoni "Movie Recommendation System", International Research Journal of Engg and Technology Volume: 05 Issue: 02, Feb-2018.
- [7] S. G Walunj, K Sadafale, "An online recommendation system for e-commerce based on Apache Mahout framework", 2017 ACM SIGMIS International Conference on Computers and People Research, pp. 153- 158,2013.
- [8] H. W. Chen, Y. L. Wu, M. K. Hor and C. Y. Tang, "Fully content-based movie recommender system with feature extraction using neural network," 2017 International Conference on Machine Learning and Cybernetics (ICMLC), Ningbo, China, 2017, pp. 504-509. doi: 10.1109/ICMLC.2017.8108968.

[9] Jain, A., &Vishwakarma, S. K., "Collaborative Filtering for Movie Recommendation using Rapid Miner" International Journal of Computer Applications (0975 - 8887) Volume 169 - No.6, July 2017.

[10] Wang, B., Xiong, S., Huang, Y., and Li, X. (2018). "Review rating prediction based on user context and product context." Applied Sciences, 8(10), 1849.

APPENDICES

A. Sample Code

HTML

```
{  
  "cells": [  
    {  
      "cell_type": "code",  
      "execution_count": 1,  
      "metadata": {},  
      "outputs": [  
        {  
          "data": {  
            "text/html": [  
              "<div>\n",  
              "<style scoped>\n",  
              "  .dataframe tbody tr th:only-of-type {\n",  
              "    vertical-align: middle;\n",  
              "  }\n",  
              "\n",  
              "  .dataframe tbody tr th {\n",  
              "    vertical-align: top;\n",  
              "  }\n",  
              "\n",  
              "  .dataframe thead th {\n",  
              "    text-align: right;\n",  
              "  }\n",  
              "</style>\n",  
            ]  
          }  
        }  
      ]  
    }  
  ]  
}
```

```

"<table border=\\"1\\" class=\\"dataframe\\">\n",
" <thead>\n",
" <tr style=\\"text-align: right;\\">\n",
" <th></th>\n",
" <th>userId</th>\n",
" <th>movieId</th>\n",
" <th>rating</th>\n",
" <th>timestamp</th>\n",
" </tr>\n",
" </thead>\n",
" <tbody>\n",
" <tr>\n",
" <td>0</td>\n",
" <td>1</td>\n",
" <td>1</td>\n",
" <td>4.0</td>\n",
" <td>964982703</td>\n",
" </tr>\n",
" <tr>\n",
" <td>1</td>\n",
" <td>1</td>\n",
" <td>3</td>\n",
" <td>4.0</td>\n",
" <td>964981247</td>\n",
" </tr>\n",
" <tr>\n",
" <td>2</td>\n",
" <td>1</td>

```

```

"    <td>6</td>\n",
"    <td>4.0</td>\n",
"    <td>964982224</td>\n",
"  </tr>\n",
"  <tr>\n",
"    <td>3</td>\n",
"    <td>1</td>\n",
"    <td>47</td>\n",
"    <td>5.0</td>\n",
"    <td>964983815</td>\n",
"  </tr>\n",
"  <tr>\n",
"    <td>4</td>\n",
"    <td>1</td>\n",
"    <td>50</td>\n",
"    <td>5.0</td>\n",
"    <td>964982931</td>\n",
"  </tr>\n",
" </tbody>\n",
"</table>\n",
"</div>"
],
"text/plain": [
"  userId  movieId  rating  timestamp\n",
"0    1    1    4.0  964982703\n",
"1    1    3    4.0  964981247\n",
"2    1    6    4.0  964982224\n",
"3    1   47    5.0  964983815\n",

```

```
"4    1    50    5.0 964982931"

]

},
```

Python

```
import numpy as np

import pandas as pd

ratings_data = pd.read_csv("ratings.csv")

ratings_data.head()

movie_names = pd.read_csv("movies.csv")

movie_names.head()

movie_data = pd.merge(ratings_data, movie_names, on='movieId')

movie_data.groupby('title')['rating'].mean().sort_values(ascending=False).head()

ratings_mean_count = pd.DataFrame(movie_data.groupby('title')['rating'].mean())

import matplotlib.pyplot as plt

import seaborn as sns

sns.set_style('dark')

%matplotlib inline

plt.figure(figsize=(8,6))

plt.rcParams['patch.force_edgecolor'] = True

ratings_mean_count['rating_counts'].hist(bins=50)

import matplotlib.pyplot as plt

import seaborn as sns

sns.set_style('dark')

%matplotlib inline
```

```
plt.figure(figsize=(8,6))
```

```
plt.rcParams['patch.force_edgecolor'] = True
```

```
ratings_mean_count['rating_counts'].hist(bins=50)
```

B) SCREEN SHOTS

```

In [8]: import numpy as np
import pandas as pd

ratings_data = pd.read_csv("ratings.csv")
ratings_data.head()

Out[8]:
   userId  movieId  rating  timestamp
0      1        1      4.0  984602703
1      1        3      4.0  984601247
2      1        6      4.0  984601228
3      1       47      5.0  984601415
4      1       50      5.0  984601931

In [9]: movie_names = pd.read_csv("movies.csv")
movie_names.head()

Out[9]:
   movieId      title  genres
0        1  Toy Story (1995)  Adventure|Animation|Children|Comedy|Fantasy
1        3  Jurassic Park (1993)  Adventure|Children|Fantasy
2        6  The Godfather Part II (1974)  Crime|Drama
3       47  The Godfather (1972)  Crime|Drama
4       50  The Godfather Part I (1972)  Crime|Drama

```

```

In [10]: movie_data = pd.merge(ratings_data, movie_names, on='movieId')

In [11]: movie_data.head()

Out[11]:
   userId  movieId  rating  timestamp      title  genres
0        1        1      4.0  984602703  Toy Story (1995)  Adventure|Animation|Children|Comedy|Fantasy
1        1        3      4.0  984601247  Jurassic Park (1993)  Adventure|Children|Fantasy
2        1        6      4.0  984601228  The Godfather Part II (1974)  Crime|Drama
3        1       47      5.0  984601415  The Godfather (1972)  Crime|Drama
4        1       50      5.0  984601931  The Godfather Part I (1972)  Crime|Drama

In [12]: movie_data.groupby('title')['rating'].mean().head()

Out[12]:
title
'11 (2004)'      4.8
'12 (2004)'      4.8
'13 (2004)'      4.8
'14 (2004)'      4.8
'15 (2004)'      4.8
Name: rating, dtype: float64

In [13]: movie_data.groupby('title')['rating'].mean().sort_values(ascending=False).head()

Out[13]:
title
The Godfather Part II (1974)      5.0
The Godfather (1972)      5.0
The Godfather Part I (1972)      5.0
The Godfather Part III (1978)      5.0
The Godfather Part IV (1983)      5.0
Name: rating, dtype: float64

```


The following movies are recommended to you

- **Spider-Man 2(Good)**
- **Spider-Man(Good)**
- **Hancock(Good)**
- **Oz: The Great and Powerful(Not Bad)**
- **The Notebook(Good)**

BACK