MOVIE RECOMMENDER SYSTEM

Project report submitted in partial fulfillment of the Requirements for the Award of the Degree of
Bachelor of Technology
In
Computer Science and Engineering

BY

NAME: AMIT RAWAT
ROLL NO. 2011302

NAME: UMANG NAITHANI
ROLL NO. 2011650

NAME: ADITYA NAUTIYAL
ROLL NO. 2011604

Department of Computer Science and Engineering

Graphic Era Deemed to be University, Dehradun, Uttarakhand(248001), June 2020

MOVIE RECOMMENDER SYSTEM

Github Link-(https://github.com/BLink291/Movie_recommendation)

Project report submitted in partial fulfillment of the Requirements for the Award of the Degree of

Bachelor of Technology
In
Computer Science and Engineering

BY

AMIT RAWAT
Roll no.-2011302

UMANG NAITHANI
Roll no.-2011650

ADITYA NAUTIYAL
Roll no.-2011604

Under the guidance of
MR. VIJAY SINGH
PROFESSOR, GEU

Department of Computer Science and Engineering

Graphic Era Deemed to be University, Dehradun, Uttarakhand(248001), June 2020

CERTIFICATE

This is to certify that the project report entitled "**MOVIE RECOMMENDER SYSTEM**" being submitted by

AMIT RAWAT	**2011302**
UMANG NAITHANI	**2011650**
ADITYA NAUTIYAL	**2011604**

in partial fulfillment for the award of the Degree of **Bachelor of Technology** in **Computer Science and Engineering** to the **Graphic Era Deemed to be University** is a record of bonafide work carried out under my guidance and supervision.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree or Diploma.

(Project Guide) Designation Date:

Head of the Department

ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of **Graphic Era Deemed to be University** , **MR. Vijay Singh** , our mentor, **Umang Naithani and Aditya Nautiyal,** my project partners . I would like to extend my sincere thanks to all of them.

I am highly indebted to **MR. Vijay Singh** for his guidance and constant supervision as well as for providing necessary information regarding the project & also for his support in completing the project.
I would like to express my gratitude towards m y parents and members of **Graphic Era Deemed to be University** for their kind cooperation and encouragement which helped me in completion of this project.
I would like to express my special gratitude and thanks to **Umang Naithani and Aditya Nautiyal,** my project partners, for being cooperative and doing his part with utmost perfection.
My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

ABSTRACT

In this project, we attempt to understand the different kinds of recommendation systems and compare their performance on the MovieLens dataset. We attempt to build a scalable model to perform this analysis. We start by preparing and comparing the various models on a smaller dataset of 100,000 ratings. Then, we try to scale the algorithm so that it is able to handle 20 million ratings by using Apache Spark. We find that for the smaller dataset, using user-based collaborative filtering results in the lowest Mean Squared Error on our dataset.

Table of Contents

CHAPTER 1: INTRODUCTION

1.1 Introduction

Rapid development of mobile devices and the internet has made it possible for us to access different movie resources freely. People sometimes find it difficult to choose from millions of movies. Moreover, movie service providers need an efficient way to manage movies and help their customers to discover movies by giving quality recommendations. Thus, there is a strong need for a good recommendation system.

Currently, there are many movie streaming services, like Amazon, Netflix, etc. which are working on building high-precision commercial movie recommendation systems. These companies generate revenue by helping their customers discover relevant movies and charging them for the quality of their recommendation service. Thus, there is a strong thriving market for good movie recommendation systems. Movie recommender system is a system which learns from the users past watching history and recommends them movies which they would probably like to watch in future. We have implemented various algorithms to try to build an effective recommender system. We firstly implemented a popularity based model which was quite simple and intuitive. Collaborative filtering algorithms which predict (filtering) taste of a user by collecting preferences and tastes from many other users (collaborating) is also implemented. We have also done experiments on content based models, based on latent factors and metadata.

1.2 Project Overview

We use machine learning to build a personalized movie scoring and recommendation system based on user's previous movie ratings. Different people have different tastes in movies, and this is not reflected in a single score that we see when we Google a movie. Our movie scoring system helps users instantly discover movies to their liking, regardless of how distinct their tastes may be.

1.3 Objectives

- System portal should have login for admins and users.
- All the users should be able to give reviews to movies.
- Admin can add movies to the database.
- Users can get recommendations of movies on popularity basis and can also filter the recommendations by genres.

1.4 Genesis of problem

The number of movies available exceeds the watching capacity of a single individual.
In the past, people used to shop in a physical store, in which the items available are limited. For instance, the number of movies that can be placed in a Blockbuster store depends on the size of that store. By contrast, nowadays, the Internet allows people to access abundant resources online. Netflix, for example, has an enormous collection of movies. Although the amount of available information increased, a new problem arose as people had a hard time selecting the items they actually wanted to see. This is where the recommender system comes in. Therefore many movie streaming services, like Amazon, Netflix, etc. are working on building high-precision commercial movie recommendation systems. These companies generate revenue by helping their customers discover relevant movies and charging them for the quality of their recommendation service. Thus, there is a strong thriving market for good movie recommendation systems.

CHAPTER 2: LITERATURE SURVEY

2.1 ALGORITHMS

For our project, we focused on two main algorithms for recommendations:

- Content-based filtering
- Collaborative filtering

2.1.1 Content Based Recommendations :

Content Based Recommendation algorithm takes into account the likes and dislikes of the user and generates a User Profile. For generating a user profile, we take into account the item profiles(vector describing an item) and their corresponding user rating. The user profile is the weighted sum of the item profiles with weights being the ratings user rated.

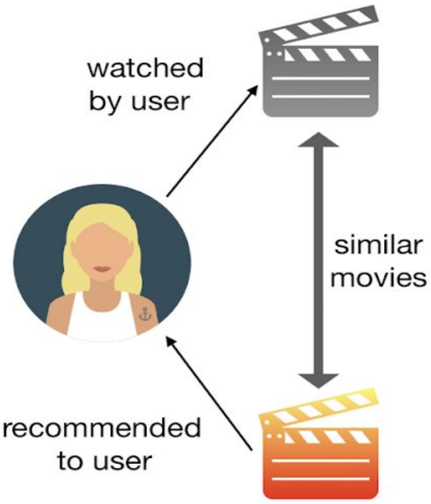


Figure 1. Content based filtering

2.1.2 Collaborative Filtering :

Collaborative Filtering techniques make recommendations for a user based on ratings and preferences data of many users. The main underlying idea is that if two users have both liked certain common items, then the items that one user has liked that the other user has not yet tried can be recommended to him. We see collaborative filtering techniques in action on various Internet platforms such as Amazon.com, Netflix, Facebook.

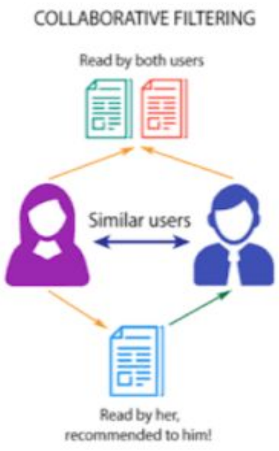
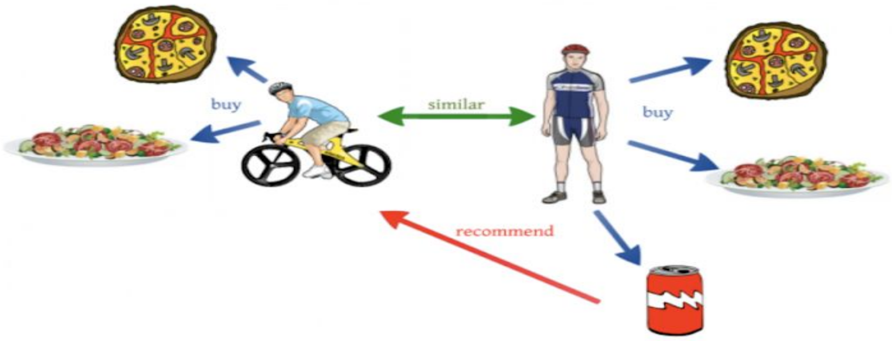


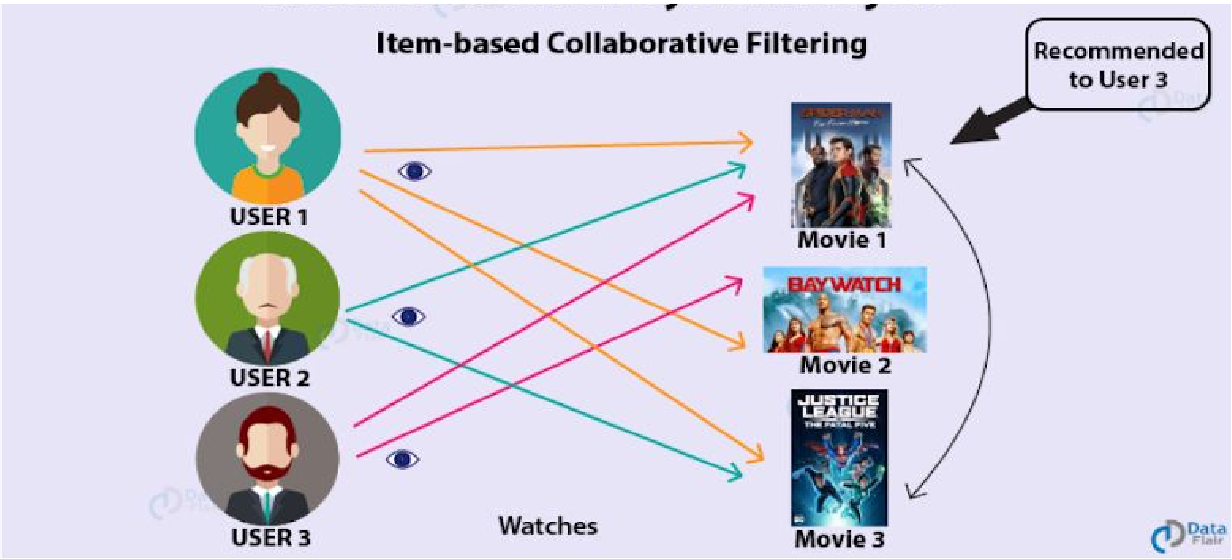
Figure 2. Collaborative Filtering

Collaborative Filtering is of two types:

User-based: For a user U, with a set of similar users determined based on rating vectors consisting of given item ratings, the rating for an item I, which hasn't been rated, is found by picking out N users from the similarity list who have rated the item I and calculating the rating based on these N ratings



Item-based: For an item I, with a set of similar items determined based on rating vectors consisting of received user ratings, the rating by a user U, who hasn't rated it, is found by picking out N items from the similarity list that have been rated by U and calculating the rating based on these N ratings.



2.2 Python

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

2.3.1 NumPy

NumPy stands for ‘Numerical Python’ or ‘Numeric Python’. It is an open source module of Python which provides fast mathematical computation on arrays and matrices. Since arrays and matrices are an essential part of the Machine Learning ecosystem, NumPy along with Machine Learning modules like Scikit-learn, Pandas, Matplotlib, TensorFlow, etc. complete the Python Machine Learning Ecosystem.

2.3.2 Pandas

Similar to NumPy, Pandas is one of the most widely used python libraries in data science. It provides high-performance, easy to use structures and data analysis tools. Unlike NumPy library which provides objects for multi-dimensional arrays, Pandas provides in-memory 2d table objects called Dataframe. It is like a spreadsheet with column names and row labels.

2.4 MongoDB

MongoDB is a cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc. and licensed under the Server Side Public License (SSPL).

2.5 HTML

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

2.6 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

2.7 JavaScript

JavaScript is a cross-platform, object-oriented scripting language used to make webpages interactive (e.g., having complex animations, clickable buttons, popup menus, etc.). There are also more advanced server side versions of JavaScript such as Node.js, which allow you to add more functionality to a website than simply downloading files (such as real time collaboration between multiple computers). Inside a host environment (for example, a web browser), JavaScript can be connected to the objects of its environment to provide programmatic control over them.

2.7 Visual Studio Code

Visual Studio Code is a free source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality. The source code is free and open-source, released under the permissive MIT License. The compiled binaries are freeware for any use.

2.8 Application Programming Interface (API)

An application programming interface (API) is a computing interface which defines interactions between multiple software intermediaries. It defines the kinds of calls or requests that can be made, how to make them, the data formats that should be used, the conventions to follow, etc. It can also provide extension mechanisms so that users can extend existing functionality in various ways and to varying degrees. An API can be entirely custom, specific to a component, or it can be designed based on an industry standard to ensure interoperability. Some APIs have to be documented, others are designed so that they can be "interrogated" to determine supported functionality. Since other components/systems rely only on the API, the system that provides the API can (ideally) change its internal details "behind" that API without affecting its users.

CHAPTER 3: SYSTEM DEVELOPMENT

3.1 Software Requirements

Client End: Internet Explorer, Google Chrome Operating System Windows/linux

Developer End: Visual studio Code, Python3, Operating System Windows/ linux

3.2 Hardware Requirements

Processor: dual core CPU

RAM: 1GB or higher

Database Server: Mongoengine for Mongodb

3.3 Functional Requirements

Register Users
Register new user
Login for existing user
Register admin

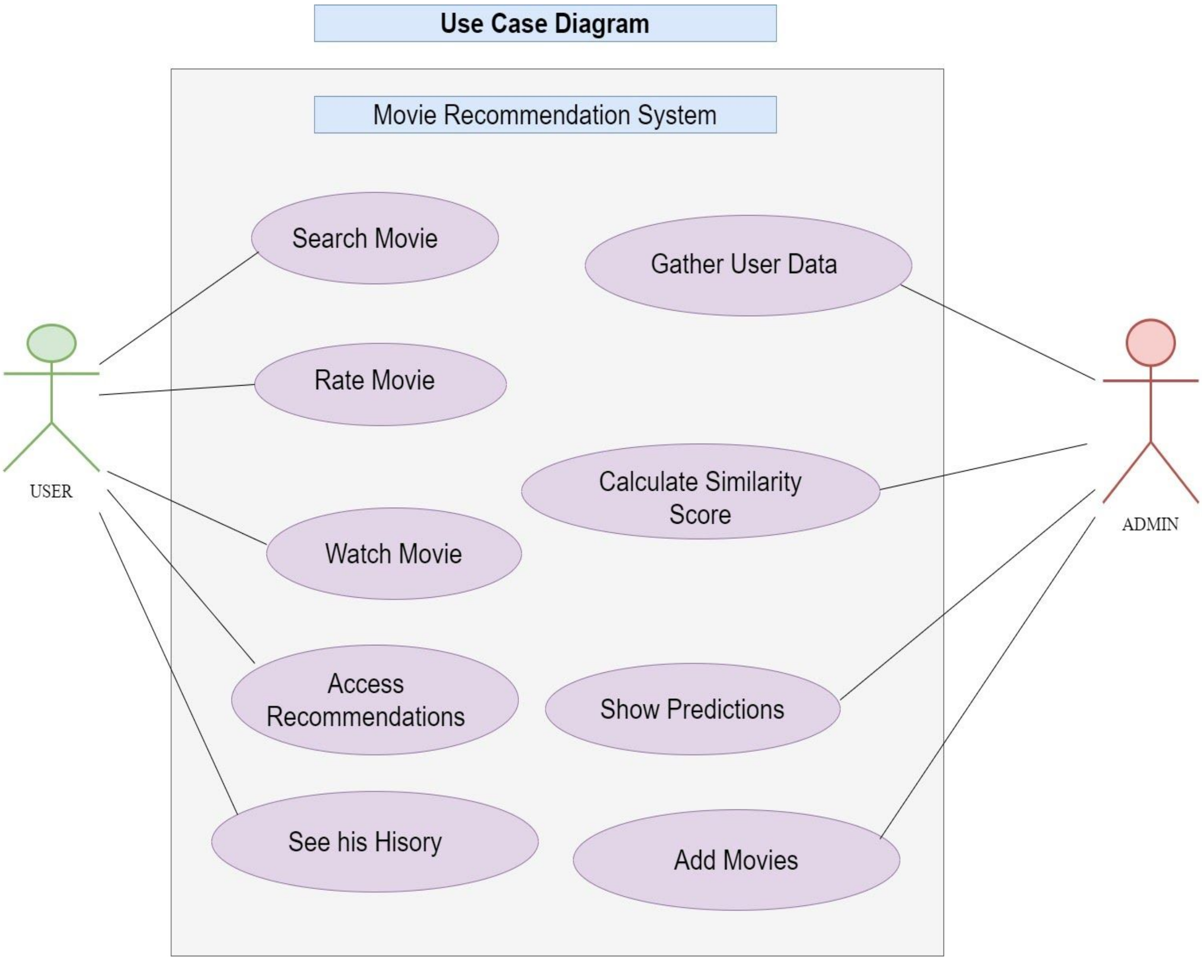
Add Movies
Admin can add movies to the database

Find Movies
Users can find any movie by their name

Review Movies
Users can give their reviews on movies

Get Recommendations
User can get personalized recommendations
User can filter the recommendations on basis of genres
User can see Top charts of movies

3.4 Use Case Diagram



3.4 User Interface (UI)

Should be user friendly
Should adapt to different screen sizes

CHAPTER 4: PROJECT DEVELOPMENT

4.1 Database Implementation

We use the MovieLens dataset available on Kaggle, covering over 45,000 movies, 26 million ratings from over 270,000 users. The data is separated into two sets: the first set consists of a list of movies with their overall ratings and features such as budget, revenue, cast, etc. After removing duplicates in the data, we have 45,433 different movies. Table 1 is the top 10 most popular movies by their weighted score, calculated using the IMDB weighting.

4.2 Tables

4.2.1 Ratings

Sl.No.	Field Name	Field Description	Field Data Type	Constraints
1.	movie_id	Unique movie Id	INTEGER	PK, NOT NULL
2.	user_id	Unique user Id	INTEGER	NOT NULL
3.	rating	Rating given by user	FLOAT	NOT NULL
4.	timestamp	Date and time of rating	DATETIME	NOT NULL

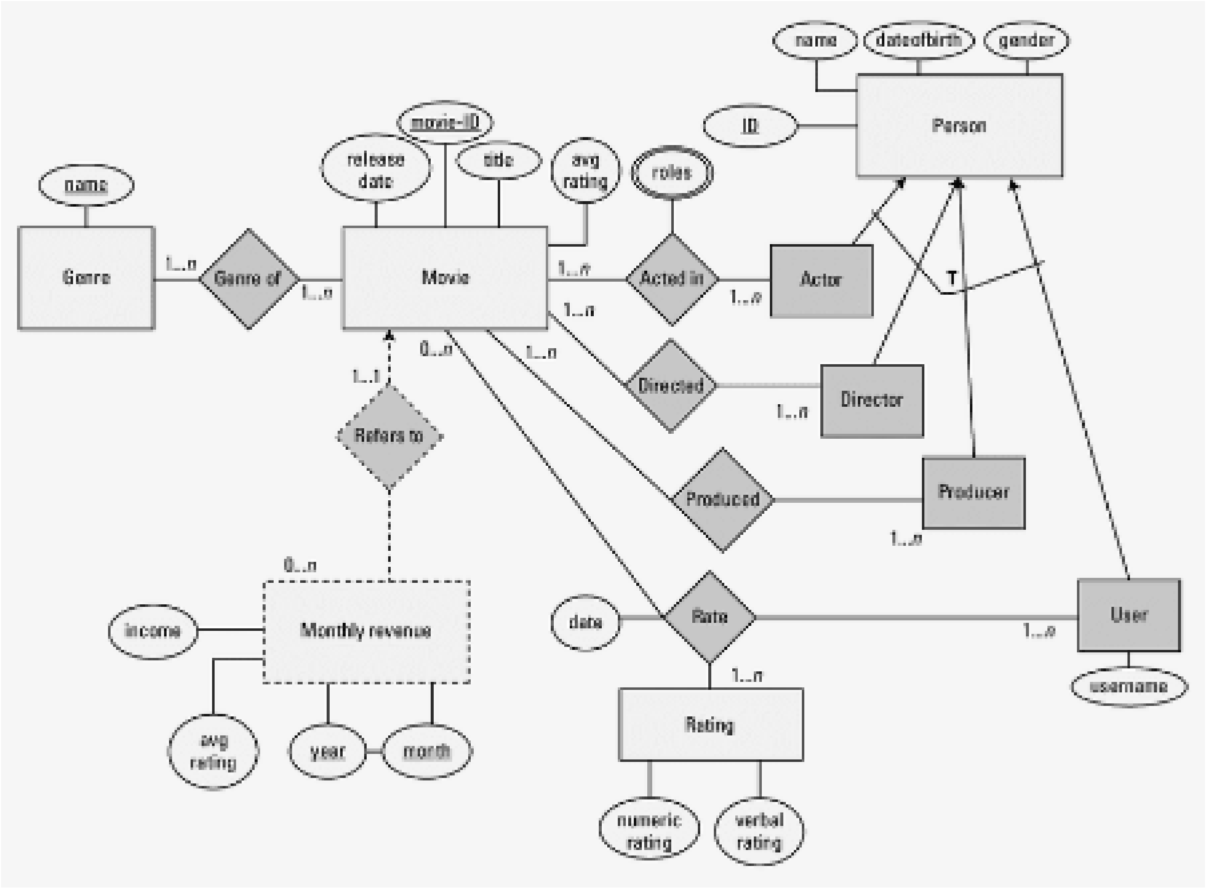
4.2.1 User

Sl.No.	Field Name	Field Description	Field Data Type	Constraints
1.	id	User Id	INTEGER	PK, NOT NULL, AUTO INCREMENT(100)
2	name	Name of user	STRING	NOT NULL
2.	email	Login Id	EMAIL FIELD	NOT NULL
3.	pswd	User Login Password	STRING	NOT NULL
4.	admin	User Creation Time	BOOLEAN	NOT NULL
5.	age	User’s age	INTEGER	--
6.	gender	User’s Gender	STRING	--
7.	country	User’s country	STRING	--
8.	spoken_languages	Languages User know	List of STRING	--

4.2.1 Movie

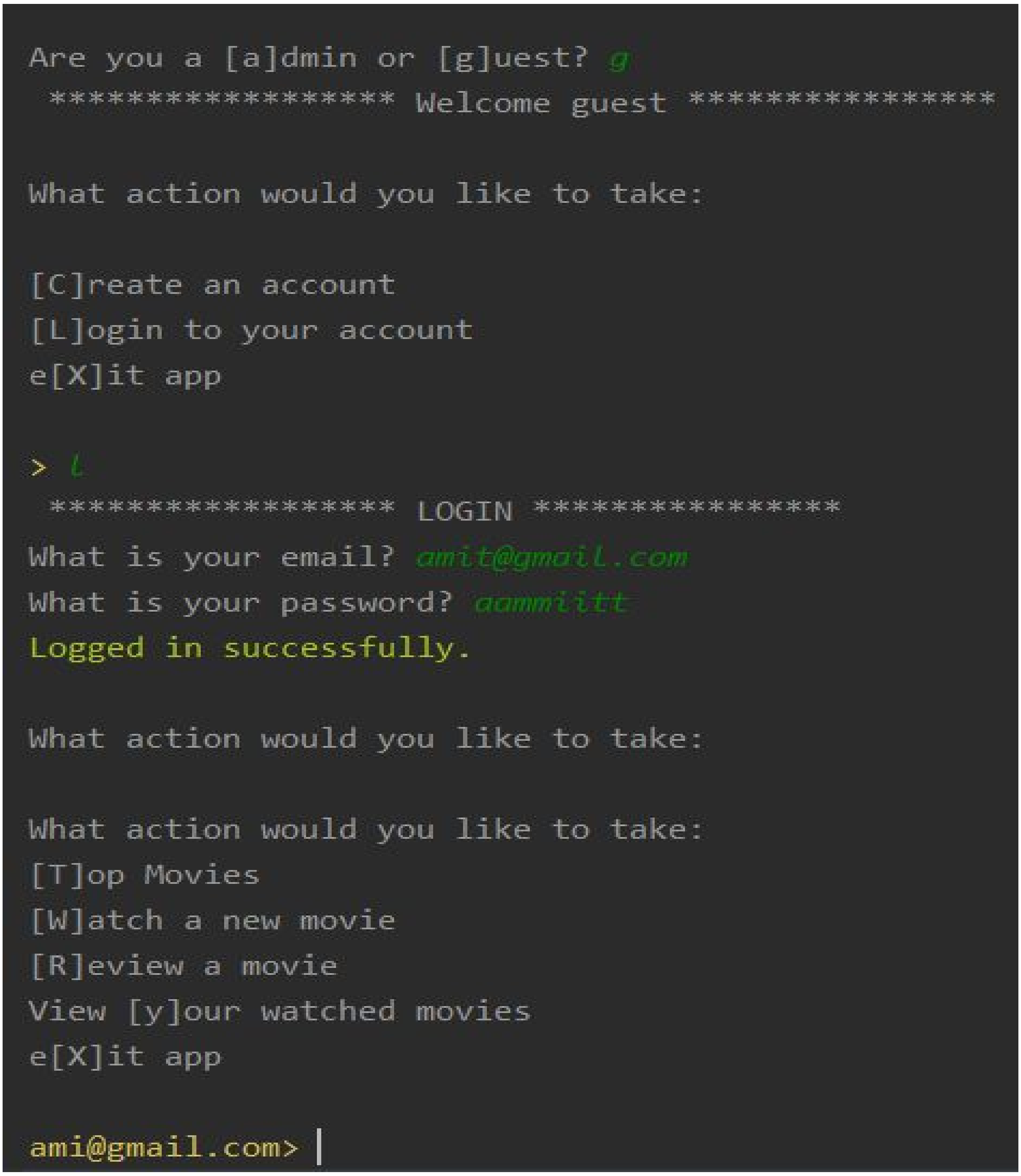
Sl.No.	Field Name	Field Description	Field Data Type	Constraints
1.	id	Movie Id	INTEGER	PK, NOT NULL, AUTO INCREMENT(100)
2	name	Name of Movie	STRING	NOT NULL
2.	genres	List of genres of movie	List of STRING	NOT NULL
3.	release _date	Release Date of the movie	STRING	NOT NULL
4.	release _year	Release Year of the movie	STRING	NOT NULL
5.	vote_count	Number of Reviews of movie	INTEGER	DEFAULT =0
6.	vote_average	Average review of movie	FLOAT	DEFAULT =0
7.	crew	List of crew involved in movie	List of STRING	--
8.	cast	List of Cast of the movie	List of STRING	--
9.	director	Director of the movie	STRING	--
10.	keywords	Keywords of that movie	List of STRING	--
11.	overview	Overview of the movie	STRING	--
12.	spoken_languages	Languages In which movie is available	List of STRING	--
13.	adult	If movie is Adult rated	BOOLEAN	DEFAULT =FALSE

4.3 ER-DIAGRAM



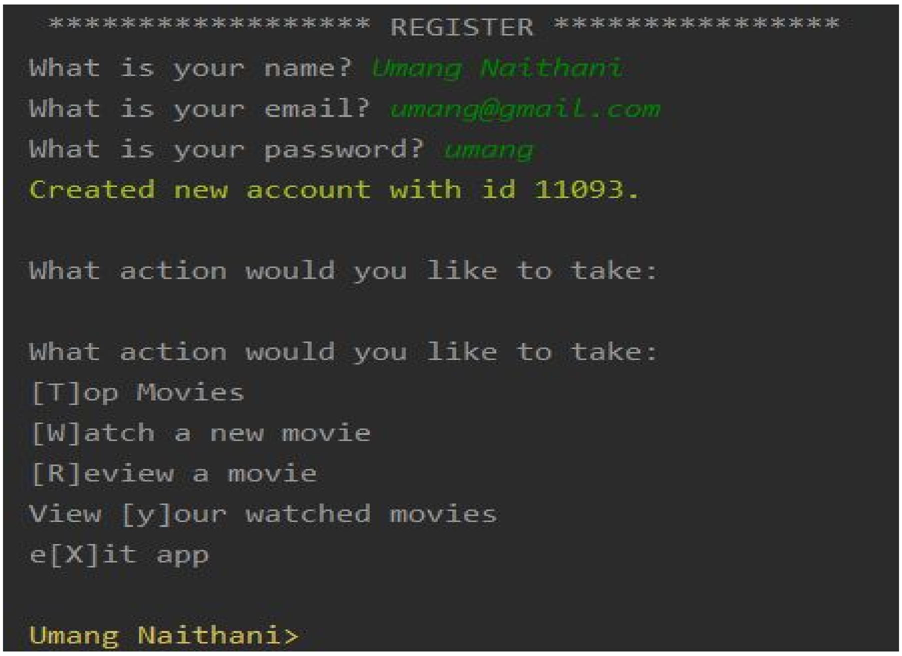
CHAPTER 5: RESULTS AND SCREENSHOTS

5.1 Login window



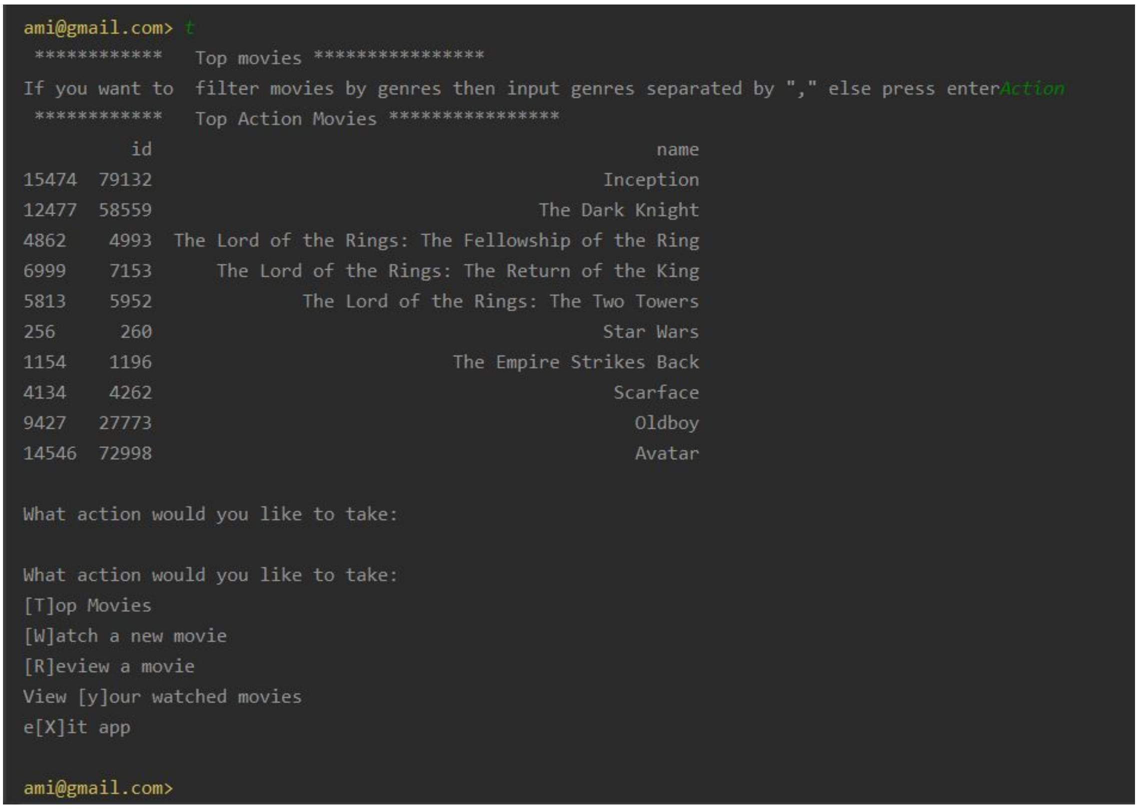
```
Are you a [a]dmin or [g]uest? g
 ****************** Welcome guest ****************

What action would you like to take:

[C]reate an account
[L]ogin to your account
e[X]it app

> l
 ****************** LOGIN ****************
What is your email? amit@gmail.com
What is your password? aammiitt
Logged in successfully.

What action would you like to take:

What action would you like to take:
[T]op Movies
[W]atch a new movie
[R]eview a movie
View [y]our watched movies
e[X]it app

ami@gmail.com> |
```

5.2 Register window



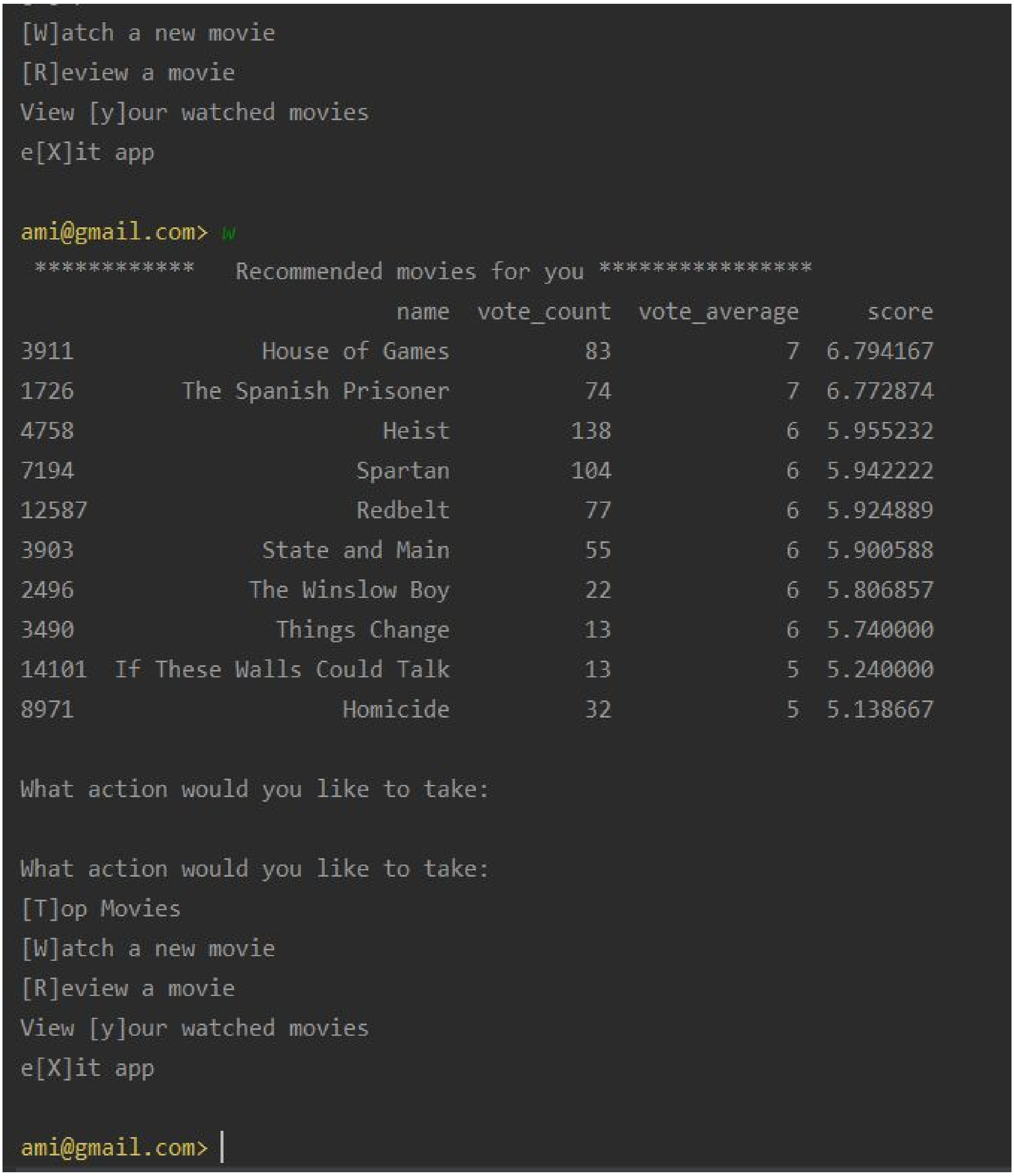
```
 ****************** REGISTER ****************
What is your name? Umang Naithani
What is your email? umang@gmail.com
What is your password? umang
Created new account with id 11093.

What action would you like to take:

What action would you like to take:
[T]op Movies
[W]atch a new movie
[R]eview a movie
View [y]our watched movies
e[X]it app

Umang Naithani>
```

5.3 See Top Charts



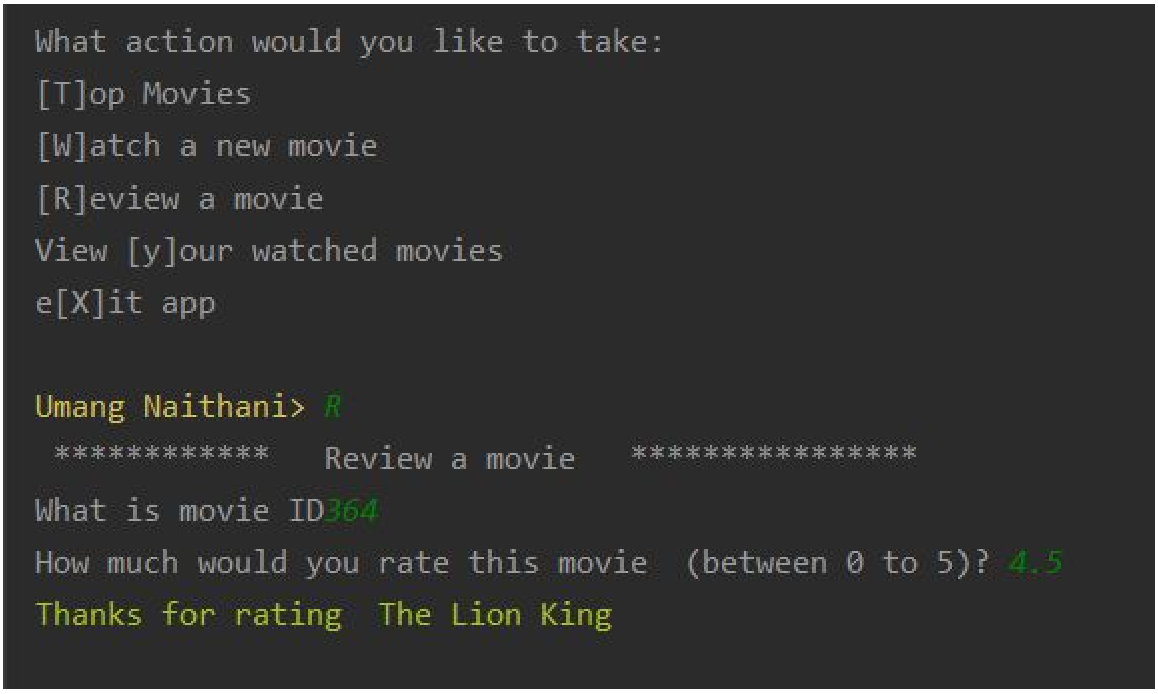
```
ami@gmail.com> t
 ************   Top movies ***************
If you want to  filter movies by genres then input genres separated by "," else press enterAction
 ************   Top Action Movies ***************
          id                                              name
15474  79132                                         Inception
12477  58559                                   The Dark Knight
4862    4993  The Lord of the Rings: The Fellowship of the Ring
6999    7153      The Lord of the Rings: The Return of the King
5813    5952              The Lord of the Rings: The Two Towers
256      260                                         Star Wars
1154    1196                           The Empire Strikes Back
4134    4262                                          Scarface
9427   27773                                           Oldboy
14546  72998                                            Avatar

What action would you like to take:

What action would you like to take:
[T]op Movies
[W]atch a new movie
[R]eview a movie
View [y]our watched movies
e[X]it app

ami@gmail.com>
```

5.4 Get Recommendation



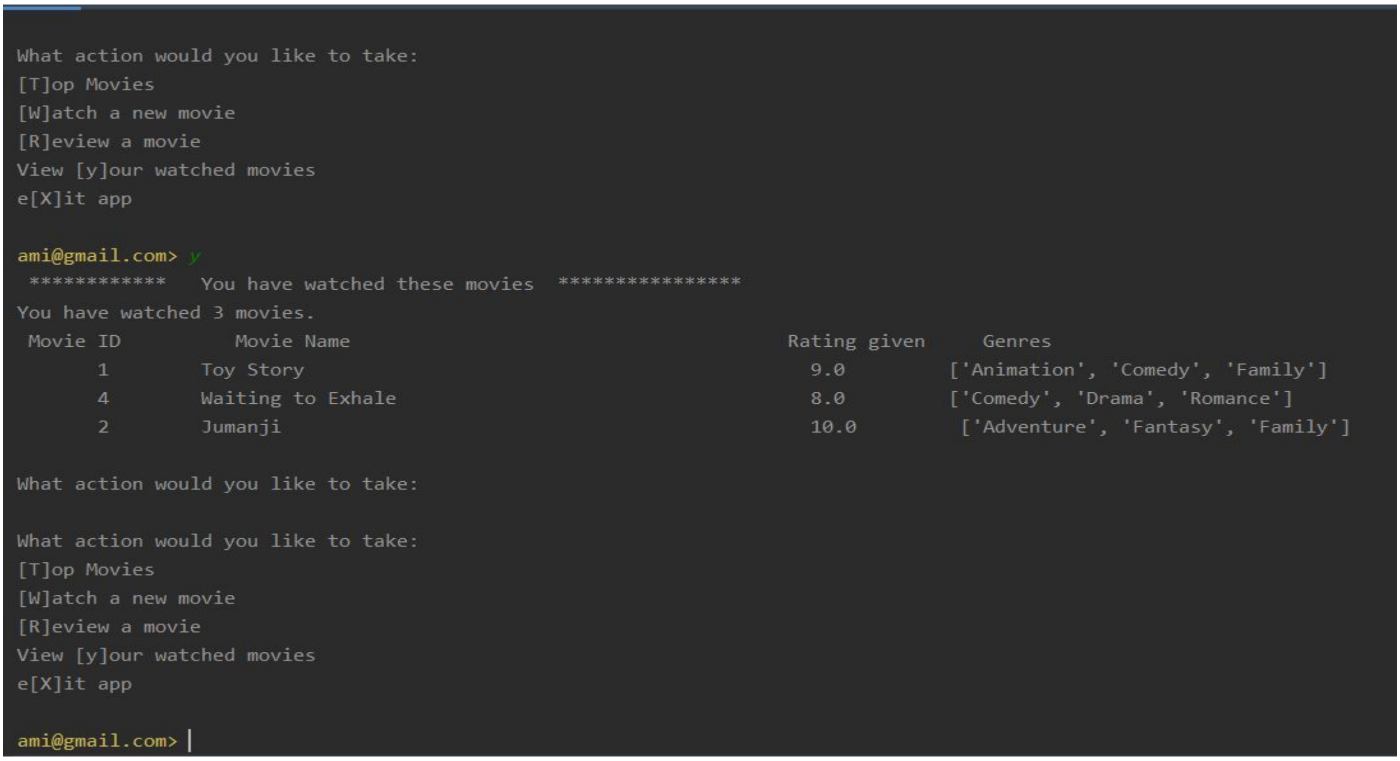
```
[W]atch a new movie
[R]eview a movie
View [y]our watched movies
e[X]it app

ami@gmail.com> w
 ************   Recommended movies for you ****************
                            name  vote_count  vote_average     score
3911              House of Games          83             7  6.794167
1726        The Spanish Prisoner          74             7  6.772874
4758                       Heist         138             6  5.955232
7194                     Spartan         104             6  5.942222
12587                    Redbelt          77             6  5.924889
3903              State and Main          55             6  5.900588
2496              The Winslow Boy         22             6  5.806857
3490               Things Change          13             6  5.740000
14101  If These Walls Could Talk          13             5  5.240000
8971                    Homicide          32             5  5.138667

What action would you like to take:

What action would you like to take:
[T]op Movies
[W]atch a new movie
[R]eview a movie
View [y]our watched movies
e[X]it app

ami@gmail.com> |
```

5.6 Review a Movie



```
What action would you like to take:
[T]op Movies
[W]atch a new movie
[R]eview a movie
View [y]our watched movies
e[X]it app

Umang Naithani> R
 ************   Review a movie   ****************
What is movie ID364
How much would you rate this movie  (between 0 to 5)? 4.5
Thanks for rating  The Lion King
```

5.7 See Users History



```
What action would you like to take:
[T]op Movies
[W]atch a new movie
[R]eview a movie
View [y]our watched movies
e[X]it app

ami@gmail.com> y
 ************   You have watched these movies   ****************
You have watched 3 movies.
 Movie ID          Movie Name                                    Rating given     Genres
       1        Toy Story                                          9.0         ['Animation', 'Comedy', 'Family']
       4        Waiting to Exhale                                  8.0         ['Comedy', 'Drama', 'Romance']
       2        Jumanji                                            10.0         ['Adventure', 'Fantasy', 'Family']

What action would you like to take:

What action would you like to take:
[T]op Movies
[W]atch a new movie
[R]eview a movie
View [y]our watched movies
e[X]it app

ami@gmail.com> |
```

CHAPTER 6: CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

In this project, a recommendation system has been implemented based on a hybrid approach of collaborative filtering engine and context based engine. The system can be highly improved by making use of caching mechanisms, user clustering which will definitely boost the speed of the system. Further enhancements include storing users past history of results, contexts for future predictions. In our project, collaborative filtering and content based filtering algorithms are used to predict a user's movie rating.

6.1 Goals Achieved

We were able to successfully build a recommendation system using a hybrid approach of both collaborative filtering and content based filtering algorithms using the MovieLens dataset available on Kaggle, covering over 45,000 movies, 26 million ratings from over 270,000 users. .

6.1 Future Scope

Run the algorithms on a distributed system, like Hadoop or Condor, to parallelize the computation, decrease the runtime and leverage distributed memory to run the complete MSD.

REFERENCES

[1] A Survey of Collaborative Filtering Techniques: Su et al; https://www.hindawi.com/journals/aai/2009/421425/

[2] Google News Personalization: Scalable Online Collaborative Filtering;

Das et al; https://www2007.org/papers/paper570.pdf

[3] Intro to Recommender Systems: Collaborative Filtering; http://blog.ethanrosenthal.com/2015/11/02/intro-to-collaborative -filtering/

[4] Movielens Dataset : https://grouplens.org/datasets/movielens/

[5] MMDS course slides; Jeffrey Ullman; http://infolab.stanford.edu/ ullman/mmds/ch9.pdf