NEW HORIZON COLLEGE OF ENGINEERING
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade. Accredited by NBA
nirf TOP ENGINEERING COLLEGE OF INDIA as per NATIONAL INSTITUTIONAL RANKING FRAMEWORK, 2018
Ministry of Human Resource Development, Government of India

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

A

PROJECT REPORT

ON

"MOVIE RECOMMENDATION USING MACHINE LEARNING "

Submitted in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

BY

SADIYA SABA (1NH16CS749)
ANUSHA S BACHIHAL (1NH16CS738)
RASHMI K (1NH16CS748)

Under the guidance of

Ms. YOGITHA
Sr. Assistant Professor,
Dept. of CSE, NHCE

NEW HORIZON COLLEGE OF ENGINEERING
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade. Accredited by NBA

nirf TOP ENGINEERING COLLEGE OF INDIA as per NATIONAL INSTITUTIONAL RANKING FRAMEWORK, 2018
Ministry of Human Resource Development, Government of India

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

It is hereby certified that the project work entitled "**MOVIE RECOMMENDATION USING MACHINE LEARNING"** is a bonafide work carried out by **SADIYA SABA (1NH16CS749) ANUSHA S BACHIHAL (1NH16CS738)** and **RASHMI K (1NH16CS748)** in partial fulfilment for the award of **Bachelor of Engineering** in **COMPUTER SCIENCE AND ENGINEERING** of the New Horizon College of Engineering during the year **2019-2020.** It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said Degree.

………………………..	………………………..	………………………………
Signature of Guide	Signature of HOD	Signature of Principal
(Ms. Anjana Sharma)	**(Dr. B. Rajalakshmi)**	**(Dr. Manjunatha)**

<u>External Viva</u>

Name of Examiner	**Signature with date**
1. ………………………………………….	**………………………………….**
2. ………………………………………….	**………………………………….**

ABSTRACT

Recommendation systems are an important part of suggesting items especially in streaming services. For streaming movie services like Netflix, recommendation systems are essential for helping users find new movies to enjoy. In this paper, we propose a deep learning approach based on auto encoders to produce a collaborative filtering system which predicts movie ratings for a user based on a large database of ratings from other users. Using the MovieLens dataset, we explore the use of deep learning to predict users' ratings on new movies, thereby enabling movie recommendations. To verify the novelty and accuracy of our deep learning approach, we compare our approach to standard collaborative filtering techniques: k-nearest- neighbour and matrix-factorization. The experimental results show that our recommendation system outperforms a user-based neighbourhood baseline both in terms of root mean squared error on predicted ratings and in a survey in which users judge between recommendations from both systems.

It recommends movies best suited for users as per their age and gender and also as per the genres they prefer to watch. The recommended movie list is created by the cumulative effect of ratings and reviews given by previous users. A neural network is trained to detect genres of movies like horror, comedy based on the emotions of the user watching the trailer. Thus, proposed system is intelligent as well as secure as a user is verified by comparing his face at the time of login with one stored at the time of registration. The system is implemented by a fully dynamic interface i.e. a website that recommends movies to the user.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be impossible without the mention of the people who made it possible, whose constant guidance and encouragement crowned our efforts with success.

I have great pleasure in expressing my deep sense of gratitude to **Dr. Mohan Manghnani**, Chairman of New Horizon Educational Institutions for providing necessary infrastructure and creating good environment.

I take this opportunity to express my profound gratitude to **Dr. Manjunatha** , Principal NHCE, for his constant support and encouragement.

I am grateful to **Dr. Prashanth C.S.R**, Dean Academics, for his unfailing encouragement and suggestions, given to me in the course of my project work.

I would also like to thank **Dr. B. Rajalakshmi**, Professor and Head, Department of Computer Science and Engineering, for her constant support.

I express my gratitude to **Ms. Yogitha,** Senior Assistant Professor, my project guide, for constantly monitoring the development of the project and setting up precise deadlines. Her valuable suggestions were the motivating factors in completing the work.

Finally a note of thanks to the teaching and non-teaching staff of Dept of Computer Science and Engineering, for their cooperation extended to me, and my friends, who helped me directly or indirectly in the course of the project work.

SADIYA SABA(1NH16CS749)

ANUSHA S BACHIHAL(1NH16CS738)

RASHMI K(1NH16CS748)

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 GENERAL INTRODUCTION

As we know that, the world is growing faster like never before. Everyone is rushing for their ultimate goals. This thirst results into the development of almost every sector. Online business is one of them. We people, don’t have time to shop from market and this is not the end. We don't even have time to choose the object from the collection. This created the embryo of online shopping, which nowadays, became a huge tree, of tons of branches.

As the online market grows exponentially, it’s obvious that competition will entered in various other fields also. Now, owners of their respective sites need to attract their users by providing attractive facilities. Recommender Engines is one of the facilities given to users. Recommender engine are the most immediately recognizable machine learning technique in use today. We will have seen services or sites that attempt to recommend books or movies or articles based on our past actions. They try to infer tastes and preferences and identify unknown items that are of interest.

Due to advances in the recommender system users constantly expect good recommendations. They have a low threshold for services that are not able to make appropriate suggestions. If a music streaming app is not able to predict and play music that the user likes, then the user will simply stop using it. This has led to a high emphasis by tech companies on improving their recommendation systems. However, the problem is more complex than it seems. Every user has different preferences like in addition even the taste of a single user can vary depending on large number of factors, such as good mood, season, or type of activity the user is doing. Since each user is different, this approach is considered to be too simple.

The basic idea behind this system is that movies that are more popular and critically acclaimed will have a higher probability of being liked by the average audience.For example, the type of music one would like to hear while exercising differs greatly from the type of music he'd listen to when cooking dinner. Netflix similarly recommends DVDs that may be of interest, and famously prize to researchers who could improve the quality of their recommendations. Social networking sites like Facebook use variants on recommender.

1.2 PROBLEM DEFINITION

This paper is based on recommendation system that recommends different things to users. This system will recommend movies to users. This system will provide more precise results as compared to the existing systems. The existing system works on individual users' rating. This may be sometime useless for the users who have different taste from the recommendations shown by the system as every user may have different tastes. This system calculates the similarities between different users and then recommend movie to them as per the ratings given by the different users of similar tastes. This will provide a precise recommendation to the user.

This is a web based as well as android system where there is a movie web service which provides services to user to rate movies, see recommendations put comments and see similar movies. There are systems which deal with the self-recommendation rather than considering the likes and dislikes of users, we thereby build a system that intakes the users wishes and then recommends a watch-list of movies which is based on their selected genre. And thus makes the watch more preferable and enjoyable to the user.

Given a set of users with their previous ratings for a set of movies, can we predict the rating they will assign to a movie they have not previously rated? Ex. "Which movie will you like" given that you have seen 'Harry Potter and the Sorcerer's Stone', 'Harry Potter and the Chamber of Secrets', 'Harry Potter and the Prisoner of Azkaban' and users who saw these movies also liked "Harry Potter and the Goblet of Fire"?

1.3 PROJECT PURPOSE

Recommender systems are information filtering tools that aspire to predict the rating for users and items, predominantly from big data to recommend their likes. Movie recommendation systems provide a mechanism to assist users in classifying users with similar interests. The purpose of a recommendation system basically is to search for content that would be interesting to an individual. Moreover, it involves a number of factors to create personalized lists of useful and interesting content specific to each user/individual.

Recommendation systems are Artificial Intelligence based algorithms that skim through all possible options and create a customized list of items that are interesting and relevant to an individual. These results are based on their profile, search/browsing history, what other people with similar traits/demographics are watching, and how likely are you to watch those movies. This is achieved through predictive modeling and heuristics with the data available.

Why recommendation system?

- Improve retention

  Caters to the user's preferences and keeps them hooked to the application.
- Increase sales

  Can improve business by a great margin by giving various recommendations of different items.
- Form habits

  Influencing usage pattern in users.
- Accelerate work

  Helps the analysts for further research and reduces their work.

1.4 PROJECT FEATURES

The basic idea behind this system is that movies that are more popular and critically acclaimed will have a higher probability of being liked by the average audience. Second is content-based filtering, where we try to profile the user's interests using information collected, and recommend items based on that profile.

- Create a user account.
- Record his/her history.
- Based on the history recommend more movies.
- Based on his previous rating, recommend movies.
- Also recommends movies based on similar genre.
- Can track the preferred IMDB rated movies based on his history.
- Can track the most preferred movie genre among n users.

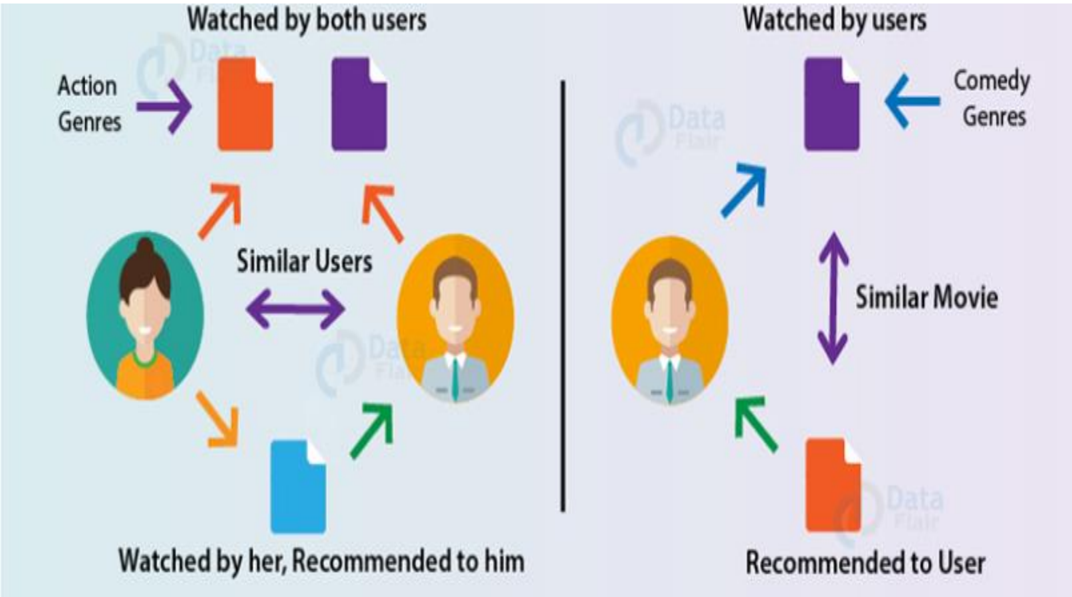


Fig 1.4 Features

1.5 MODULE DESCRIPTION

A. Admin

The system admin will add movie in a database, view movies and update it.

B. Recommendation Engine

This recommendation engine will calculate the similarities between the different users. On the basis of that similarities calculated, this engine will recommend movie to a user.

C. Movie Web Service

This will allow user to rate movies, comments on movies. This service will also show the movie recommendation to the users.

D. User

The android user can rate a movie, can comment on any movie, and can see similar movies recommended by other users who are similar to this user.

CHAPTER 2

LITERATURE SURVEY

2.1 BACKGROUND

Over the past decade, a large number of recommendation systems for a variety of domains have been developed and are in use. These recommendation systems use a variety of methods such as content based approach, collaborative approach, knowledge based approach, utility based approach, hybrid approach, etc. Most of the online recommendation systems for a variety of items use ratings from previous users to make recommendations to current users with similar interests. One such system was designed by Jung, Harris, Webster and Herlocker (2004) for improving search results. The system encourages users to enter longer and more informative search queries, and collects ratings from users as to whether search results meet their information need or not. These ratings are then used to make recommendations to later users with similar needs.

2.2 EXISTING PRODUCTS AND SYSTEMS

Current Social Networking World Internet social networking sites, which began in 1995 with Classmates.com, have surged in popularity and use through word-of-mouth advertising. Since then, a wide range of virtual communities have formed serving different purposes and targeting varying niche audiences:

In particular, we've chosen to explore the movie niche as this is an area where our project can significant improvements compared to existing products and systems. Traditional movie websites (IMDB, AOL Movies) function by proving global user ratings on movies in their database. Movies are categorized by metadata such as genre, era, directors, and so on. Users can search for movies, browse lists and read reviews written by critics or other users. However, most of these services lack any personal recommendation system and haven't taken advantage of social-networking communities or crowd wisdom. Some websites, such as

Blockbuster, do provide individualized recommendations based on a user's ratings but donot include any social networking component. Yahoo! Movies goes further and uses personal ratings to suggest movies currently playing in theatre, on TV, and out on DVD. It also draws upon its vast user base to give lists of similar movie fans, their ratings, and reviews. Other movie sites, like Flixster, take a different approach. Flixster forms web-based communities around movies and suggests movies to watch based on what your friends have rated.

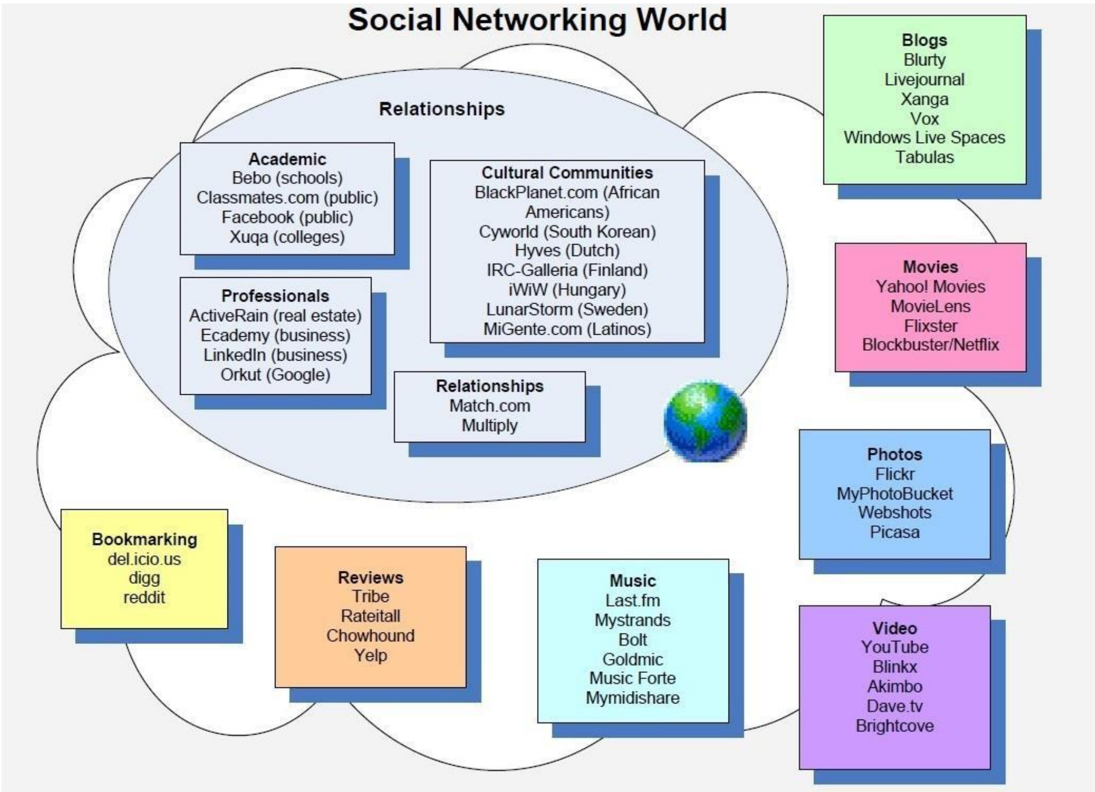


Fig 2.2 Existing System

2.3 PROPOSED SYSTEM

The system is built on windows 2007 operating system. The system uses advanced java technology along with machine learning concepts. MySQL is used for storing data. This system uses three-tier architecture. The web service layer provides the android user to rate movies, view similar recommendations given by the system and comment on it.

The proposed system is a better system than any other existing systems. This system has added the positive features of existing systems and has overcome the drawbacks of existing systems. The current system can:

- Create a user account.
- Record his/her history.
- Based on the history recommend more movies.
- Based on his previous rating, recommend movies.
- Also recommends movies based on similar genre.
- Can track the preferred IMDB rated movies based on his history.
- Can track the most preferred movie genre among n users.

The system uses all the existing algorithms i.e. content based, context based and collaborative based algorithms. All these algorithms are combined to give more precise result. We here use the Apriori algorithm to make this possible. Apriori algorithm is used for finding frequent item-sets in a dataset for Boolean association rule. Name of the algorithm is Apriori because it uses prior knowledge of frequent item-sets properties. Apriori algorithm is applied when there are several items of relationships that want to be analyzed.

Testing data mining processing then from the results of product output recommendations appropriate for customers will be done gradually by looking at the next customer purchase history data in a certain period. Apriori property which helps by reducing the search space.

The following modules are developed as:

A. Admin

The system admin will add movie in a database, view movies and update it. Keeps the tab of the user history too.

B. Recommendation Engine

This recommendation engine will calculate the similarities between the different users. On the basis of that similarities calculated, this engine will recommend movie to a user.

Based on the selected genre movies are recommended and the selected genre is stored for further recommendations.

C. Movie Web Service

This will allow user to rate movies, comments on movies. This service will also show the movie recommendation to the users.

D. Android User

The android user can rate a movie, can comment on any movie, and can see similar movies recommended by other users who are similar to this user. Can view the graph on his watch history and find the most preferred genre.

2.4 SYSTEM STUDY

FEASIBILITY 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

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.

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.

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.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FUNCTIONAL REQUIREMENTS:

Client application:

The client application is the link between the user and the server application. Its task is to gather information from the users and to allow users to play movies. The information is sent to the server application, where it is stored, and later used to produce recommendations.

In addition, the information is used to measure recommender precision. This allows for investigation of how precision is influenced by different recommender strategies.

The requirements for the client application are:

R0 - Play movies:

The client application shall provide an interface that makes it possible to play movies by selection, or by navigation through standard movie player buttons like play, pause, stop and skip.

R1 - Request recommendations:

The client application shall make it possible to request recommendations and to send the requests to the server application.

R2 - Evaluate movies:

The client application shall make it possible to evaluate each movie and to send this information to the server application. Can also rate it, view it, thus all the user activity is stored as such for further references.

Server application:

Application server frameworks are software frameworks for building application servers. An application server framework provides both facilities to create web applications and a server environment to run them. The server application receives information from the client application, and provides the client application with recommendations.

The requirements for the server application are:

R3 - Handle recommendation requests:

The server application shall receive and handle requests for recommendations.

R4 - Store evaluations:

The server application shall receive and store movie evaluations.

R5 - Recommend using Apriori filtering:

The server application shall be capable of producing recommendations by interpreting contextual information given by the users, and evaluations given by the actual user and other similar users. Gives out the frequent movies selected. We here store the generated output in the class objects for future concerns. This algorithm is for the item set based filtration.

3.2 Non-functional requirements

Non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behavior or functions. The plan for implementing functional requirements is detailed in the system design.

R8 - Accuracy The server application:

Shall produce accurate recommendations that match the user's movie preference.

R9 – Intrusiveness:

The client application shall minimize intrusiveness and at the same time capture user attention so that an acceptable amount of evaluation data is received.

R10 - Scale potential:

The recommender system shall have the potential of being scalable both with respect to size and geography

Some Non-Functional Requirements are as follows:

- Reliability
- Maintainability
- Performance
- Portability
- Scalability
- Flexibility

3.2.1 ACCESSIBILITY:

Accessibility is a general term used to describe the degree to which a product, device, service, or environment is accessible by as many people as possible.

In my project, students and faculties can login from any system as this is a desktop application, it is not system dependent. User interface is simple and efficient and easy to use.

3.2.2 MAINTAINABILITY:

In software engineering, maintainability is the ease with which a software product can be modified in order to:

- Correct defects
- Meet new requirements

New functionalities can be added in the project based on the user requirements.

Since the programming is very simple, it is easier to find and correct the defects and to make the changes in the project.

3.2.3 SCALABILITY:

System is capable of handling increase total throughput under an increased load when resources (typically hardware) are added. System can work normally under situations such as low bandwidth and large number of users.

3.2.4 PORTABILITY:

Portability is one of the key concepts of high-level programming. Portability is the software code base feature to be able to reuse the existing code instead of creating new code when moving software from an environment to another. Project can be executed under different operation conditions provided it meets its minimum configurations. Only system files and dependant assemblies would have to be configured in such case.

3.2.5 RELAIBILITY:

Software Reliability is the probability of failure-free software operation for a specified period of time in a specified environment. Software Reliability is also an important factor affecting system reliability.

3.3 HARDWARE REQUIREMENTS

- **System** : Pentium IV 2.4 GHz.
- **Hard Disk** : 40 GB.
- **Floppy Drive** : 1.44 Mb.
- **Monitor** : 14’ Colour Monitor.
- **Mouse** : Optical Mouse.
- **Ram** : 512 Mb.

3.4 SOFTWARE REQUIREMENTS

- **Operating system** : windows 7 Ultimate
- **Coding language** : Python
- **Front end** : Python

3.5 SOFTWARE DESCRIPTION

PYTHON

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++or Java. It provides constructs that enable clear programming on both small and large scales. Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations.

Python is managed by the non-profit Python Software Foundation. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

DATASET

A data set (or dataset) is a collection of data. In the case of tabular data, a data set corresponds to one or more database tables, where every column of a table represents a particular variable, and each row corresponds to a given record of the data set in question.

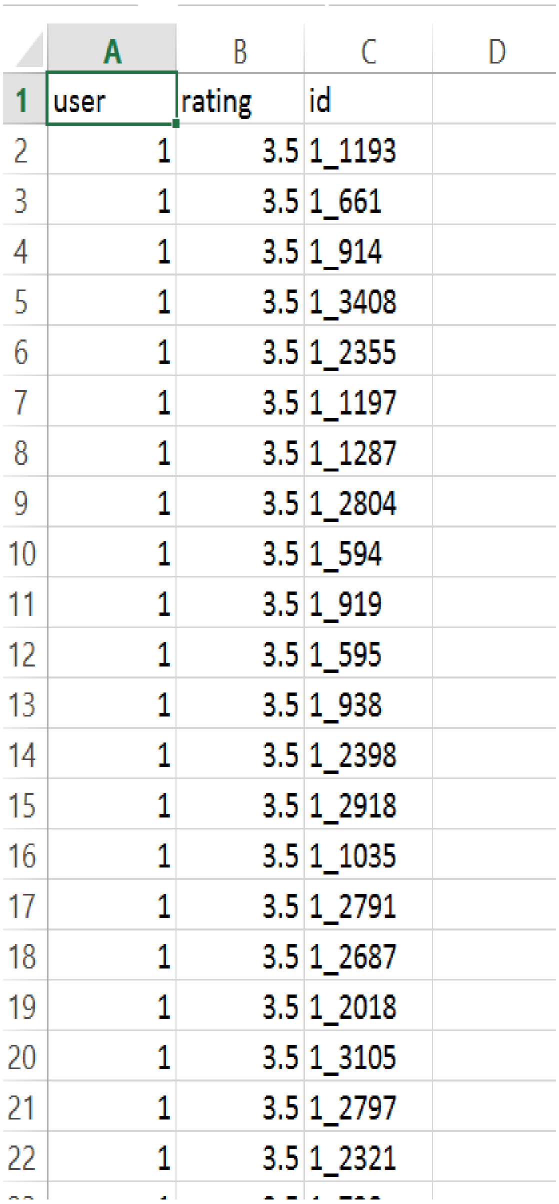
For the project we have utilized the movie dataset from 'Movie Lens(Kaggle)'.

- Movie Lens is a data set that provides 10000054 user ratings on movies.
- 95580 tags applied to 10681 movies by 71567 users.
- Users of Movie Lens were selected randomly.
- All users rated at least 20 movies.

• Each user represented by a unique id.

• u.info : The number of users, items, and ratings in the u data ssset.

• u.item : Information about the items



	A	B	C	D
1	user	rating	id	
2	1	3.5	1_1193	
3	1	3.5	1_661	
4	1	3.5	1_914	
5	1	3.5	1_3408	
6	1	3.5	1_2355	
7	1	3.5	1_1197	
8	1	3.5	1_1287	
9	1	3.5	1_2804	
10	1	3.5	1_594	
11	1	3.5	1_919	
12	1	3.5	1_595	
13	1	3.5	1_938	
14	1	3.5	1_2398	
15	1	3.5	1_2918	
16	1	3.5	1_1035	
17	1	3.5	1_2791	
18	1	3.5	1_2687	
19	1	3.5	1_2018	
20	1	3.5	1_3105	
21	1	3.5	1_2797	
22	1	3.5	1_2321	

Fig 3.5.1 Dataset(a)

movie id | movie title | release date |

• u.genre : A list of the Action

Adventure | Animation | Children's | Comedy | Crime | Documentary | Drama | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western genre genres.

```
1::Toy Story (1995)::Animation|Children's|Comedy
2::Jumanji (1995)::Adventure|Children's|Fantasy
3::Grumpier Old Men (1995)::Comedy|Romance
4::Waiting to Exhale (1995)::Comedy|Drama
5::Father of the Bride Part II (1995)::Comedy
6::Heat (1995)::Action|Crime|Thriller
7::Sabrina (1995)::Comedy|Romance
8::Tom and Huck (1995)::Adventure|Children's
9::Sudden Death (1995)::Action
10::GoldenEye (1995)::Action|Adventure|Thriller
11::American President, The (1995)::Comedy|Drama|Romance
12::Dracula: Dead and Loving It (1995)::Comedy|Horror
13::Balto (1995)::Animation|Children's
14::Nixon (1995)::Drama
15::Cutthroat Island (1995)::Action|Adventure|Romance
16::Casino (1995)::Drama|Thriller
17::Sense and Sensibility (1995)::Drama|Romance
18::Four Rooms (1995)::Thriller
19::Ace Ventura: When Nature Calls (1995)::Comedy
20::Money Train (1995)::Action
21::Get Shorty (1995)::Action|Comedy|Drama
22::Copycat (1995)::Crime|Drama|Thriller
23::Assassins (1995)::Thriller
24::Powder (1995)::Drama|Sci-Fi
25::Leaving Las Vegas (1995)::Drama|Romance
26::Othello (1995)::Drama
27::Now and Then (1995)::Drama
28::Persuasion (1995)::Romance
29::City of Lost Children, The (1995)::Adventure|Sci-Fi
```

Fig 3.5.2 Dataset (b)

u.user : Information about the users

user id | age | gender | occupation | zip code

	A	B	C	D
1	user	rating	id	
2	1	3.5	1_1193	
3	1	3.5	1_661	
4	1	3.5	1_914	
5	1	3.5	1_3408	
6	1	3.5	1_2355	
7	1	3.5	1_1197	
8	1	3.5	1_1287	
9	1	3.5	1_2804	
10	1	3.5	1_594	
11	1	3.5	1_919	
12	1	3.5	1_595	
13	1	3.5	1_938	
14	1	3.5	1_2398	
15	1	3.5	1_2918	
16	1	3.5	1_1035	
17	1	3.5	1_2791	
18	1	3.5	1_2687	
19	1	3.5	1_2018	
20	1	3.5	1_3105	
21	1	3.5	1_2797	
22	1	3.5	1_2321	

Fig 3.5.3 Dataset(c)

Load data set

In the modeling part of the model, the dataset can be used this s known as multi-class classification problem. Four features are included in the data set movie name, IMDB rating, language, country, year of release.

Split set data

To understand model performance, it is a good strategy to divide the data set into training and test sets.Let's split the data set using the function train_test_split(). Functions, target and test_set size of 3 parameters needed.

CHAPTER 4

DESIGN

4.1 SYSTEM ARCHITECTURE

System architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

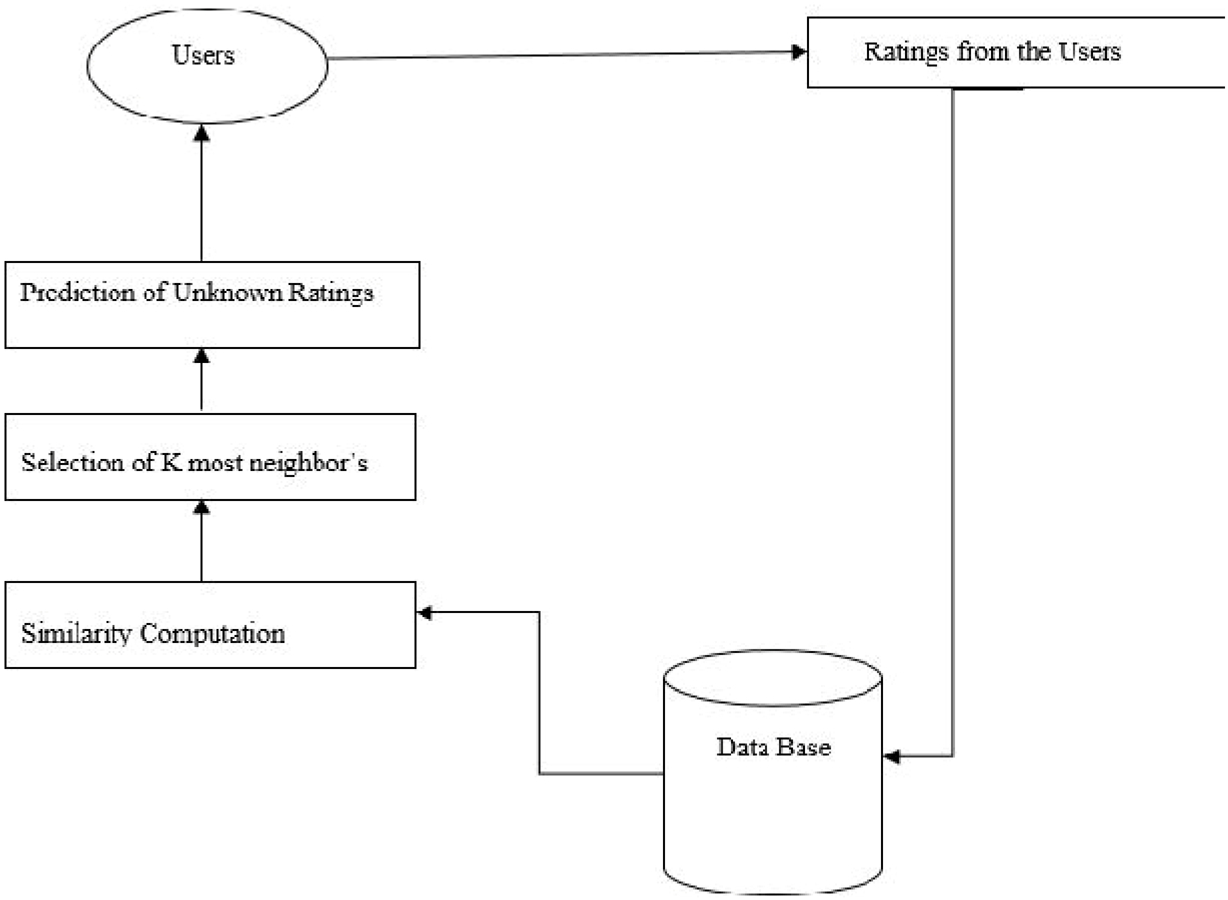


Fig 4.1 System architecture

4.1.1 CLIENT AND SERVER INTERACTION

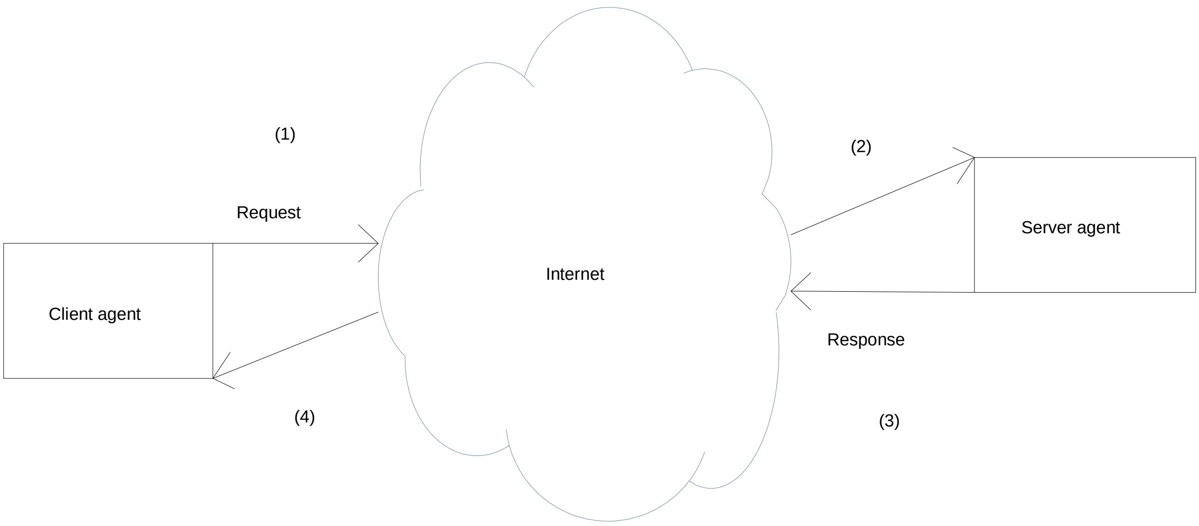


Fig 4.1.1 client server interaction

- When a request is made through the user through the application, the request therefore the request is then accepted by the server.
- Server acts as a medium between the user's interest and his request.
- The server then accepts it through the ip and the user is granted their recommendations.

4.1.2 INFORMATION FILTERING IN RECOMMENDATION SYSTEM

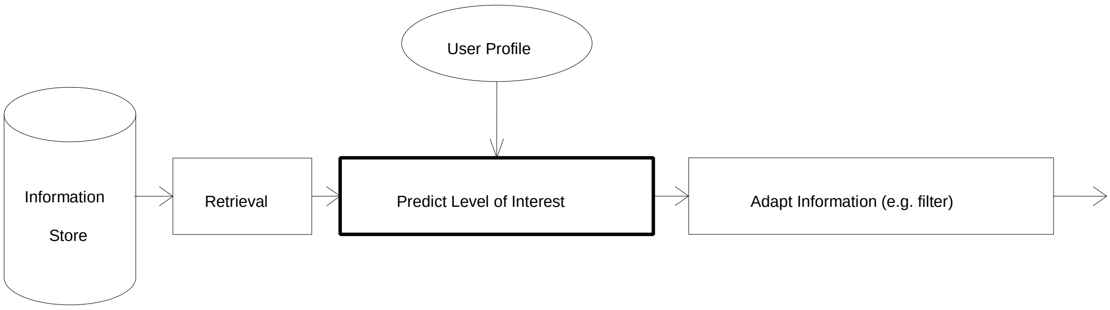


Fig 4.1.2 Information filtering in recommendation

4.3 USE CASE DIAGRAM

- Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform.
- First function is to give input i.e. set of movies with their details like movie name, language and to which genre it belongs.
- Apriori is an algorithm for frequent item set mining and association rule learning over relational databases. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database.
- Then recommends the movie based on Content-based Recommendation and

  Collaborative Filtering Algorithm.
- Based on the suggestion given by the system user will choose the movie.

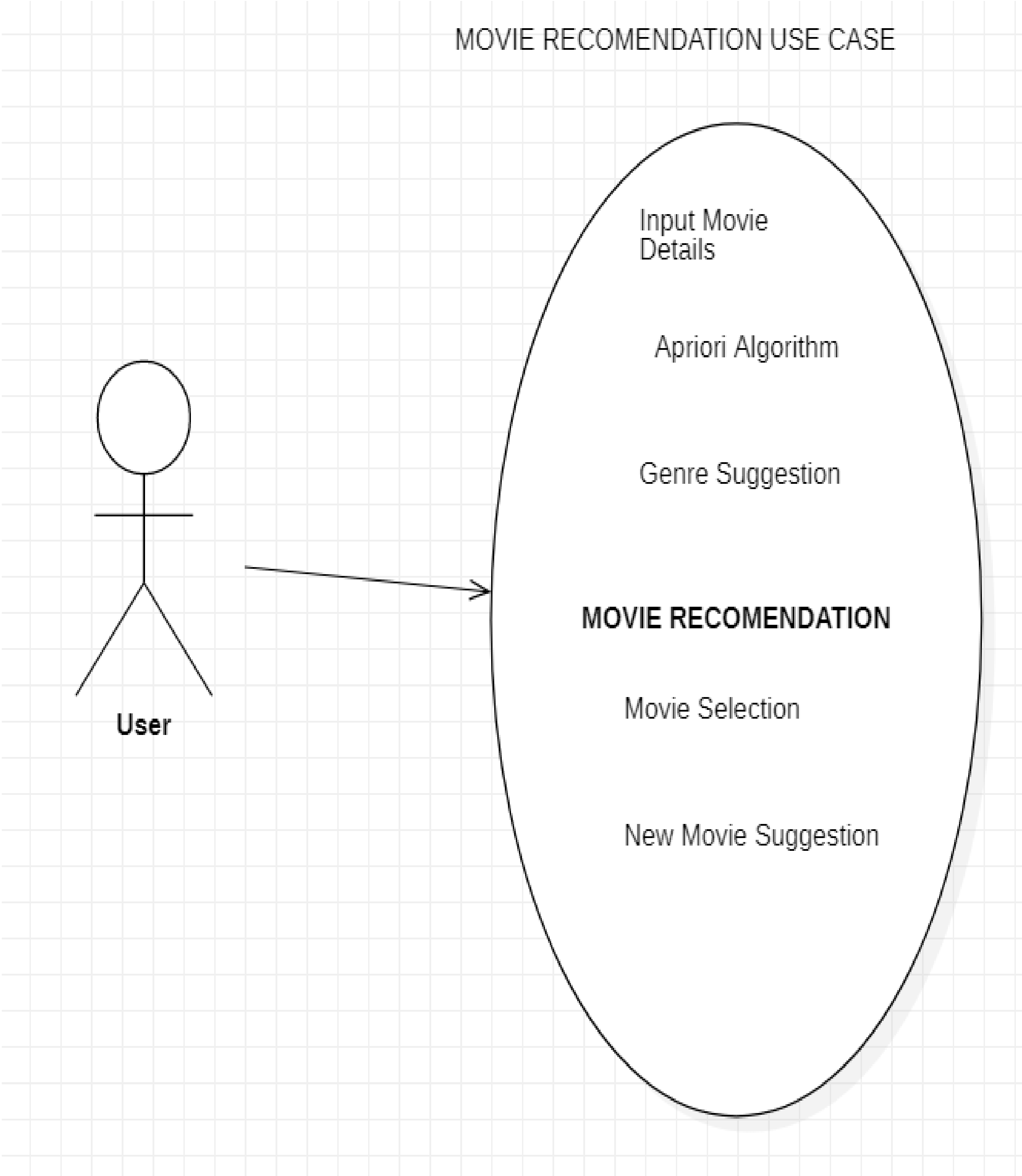


Fig 4.3 Use case diagram

4.4 STATE DIAGRAM

- A state diagram is a diagram used in computer science to describe the behavior of a system considering all the possible states of an object when an event occurs. State diagrams graphically represent finite state machines. They are only used to understand object behavior throughout the whole system.
- In the above diagram it describes the behavior of movie recommendation system.
- When the data of movie is given it checks for the similar genre by using recommendation algorithm and suggests the movie to user.
- Suggestion is based on the movie ratings, genre, language and frequently watched movies.
- If the previously watched movie’s rating is three, then the suggested movie’s rating should be equal or greater than three.

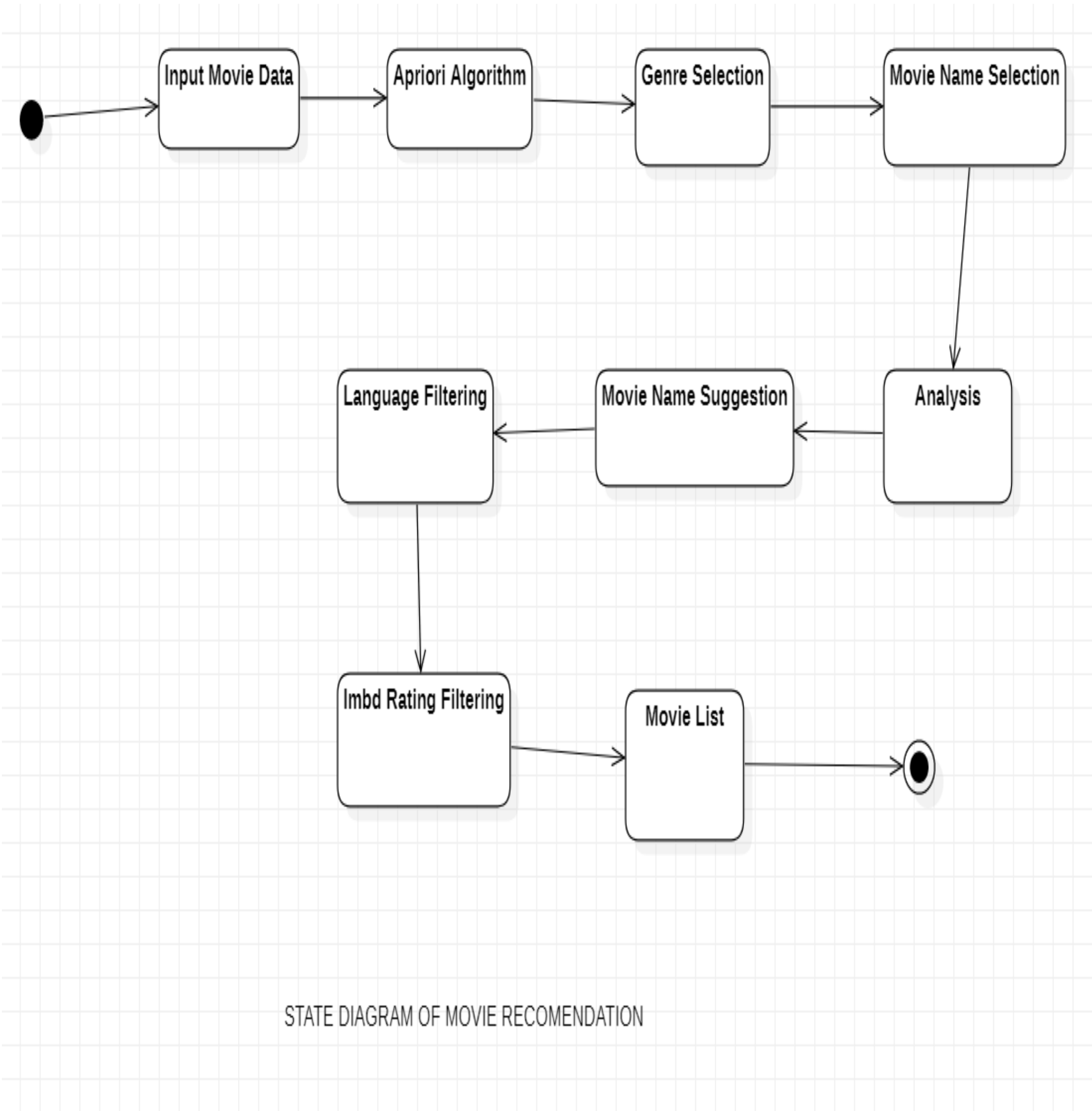


Fig 4.4 State diagram

4.5 CLASS DIAGRAM

- Class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.
- The above class diagram describes the classes, attributes and methods of movie recommendation system.
- Input movie data: It is class which takes the data of the movie details.
- Apriori algorithm: This class will do analyses based on the given data and suggests movie.
- Movie selection: This class helps in selecting movie based on the genre and the movie ratings.
- Movie suggestion: This class suggests the movie to user based on the frequently watched movie(genre).

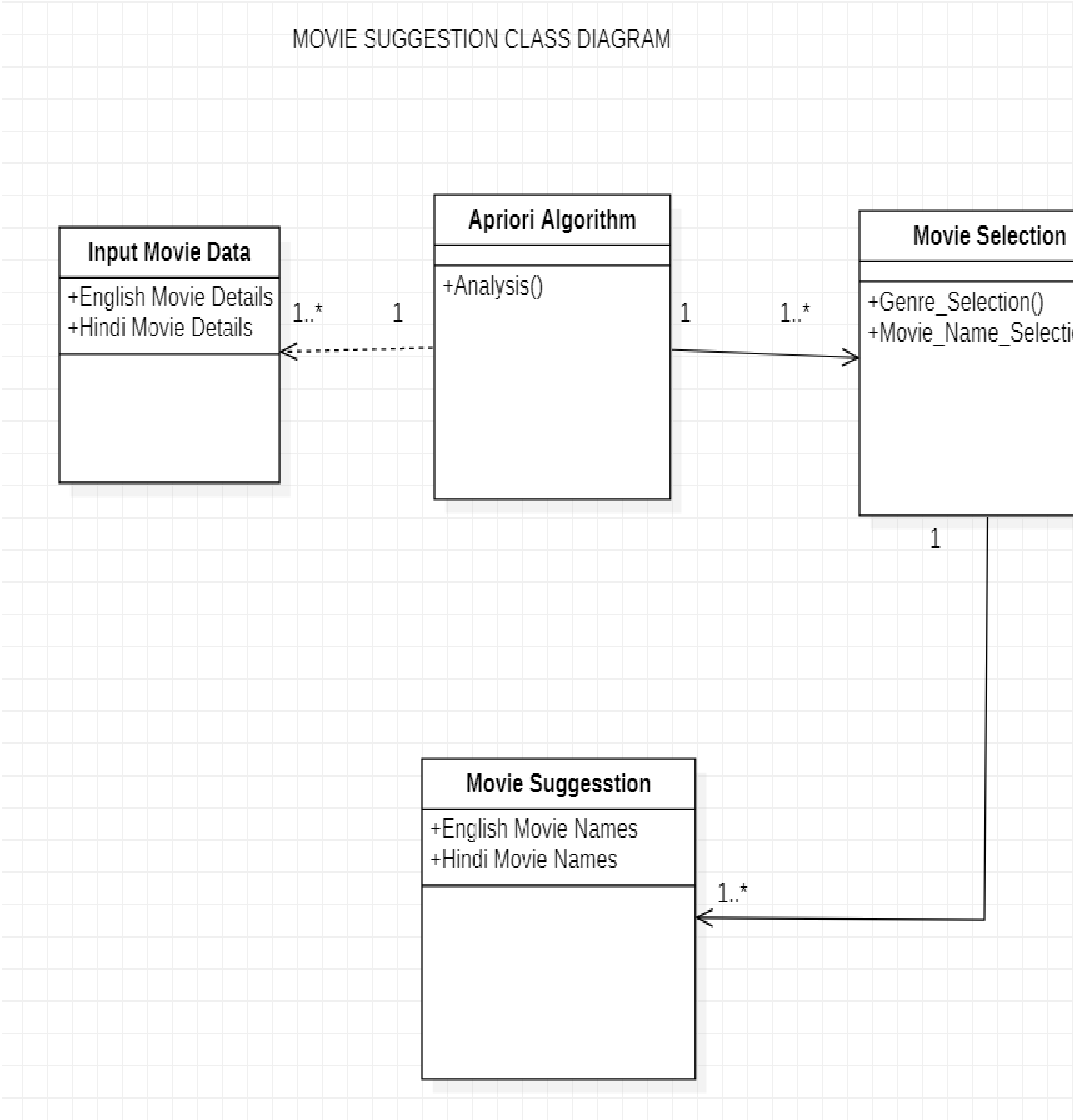


Fig 4.5 Class diagram

4.6 ACTIVITY DIAGRAM

- Activity diagram is defined as a UML diagram that focuses on the execution and flow of the behavior of a system instead of implementation. It is also called object-oriented flowchart. Activity diagrams consist of activities that are made up of actions which apply to behavioral modeling technology.
- Firstly, have to enter the user credentials.
- If a new user has to register.
- Then select the applicable genre and the movies are recommended.
- For this the system uses the Apriori algorithm, and thus the process continues and always a new recommendation list is generated.

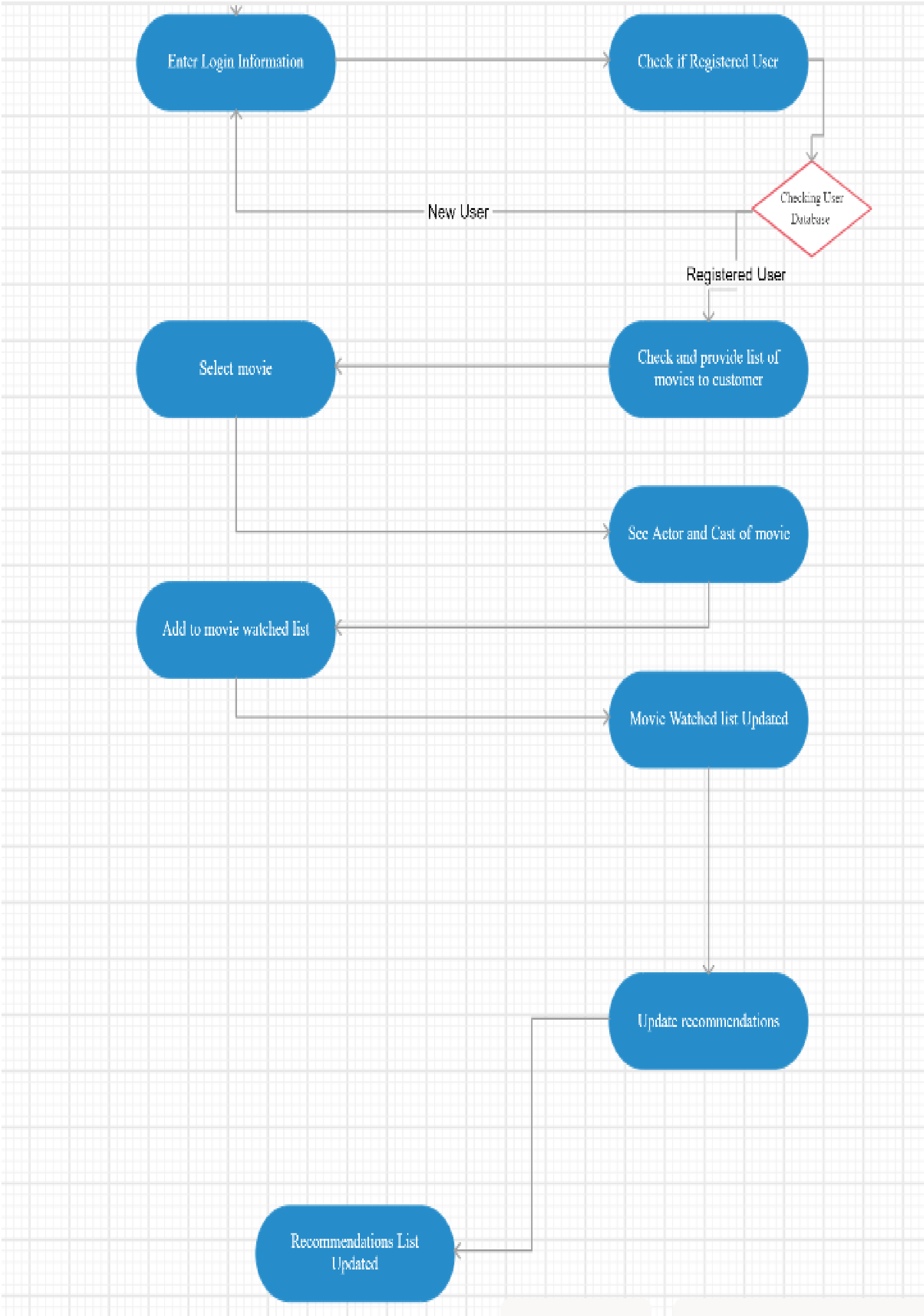


Fig 4.6 Activity diagram

4.8 SEQUENCE DIAGRAM

- A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function.
- Once a movie is selected by the user the server gets the details of his previously watched movies.
- The information is stored and if we want to see the information again we can just check for the user history using the userid.
- For further the recommendation can be shared with friends on different platforms.
- The generation of recommendations goes on in loop until the user selects one for his watch.

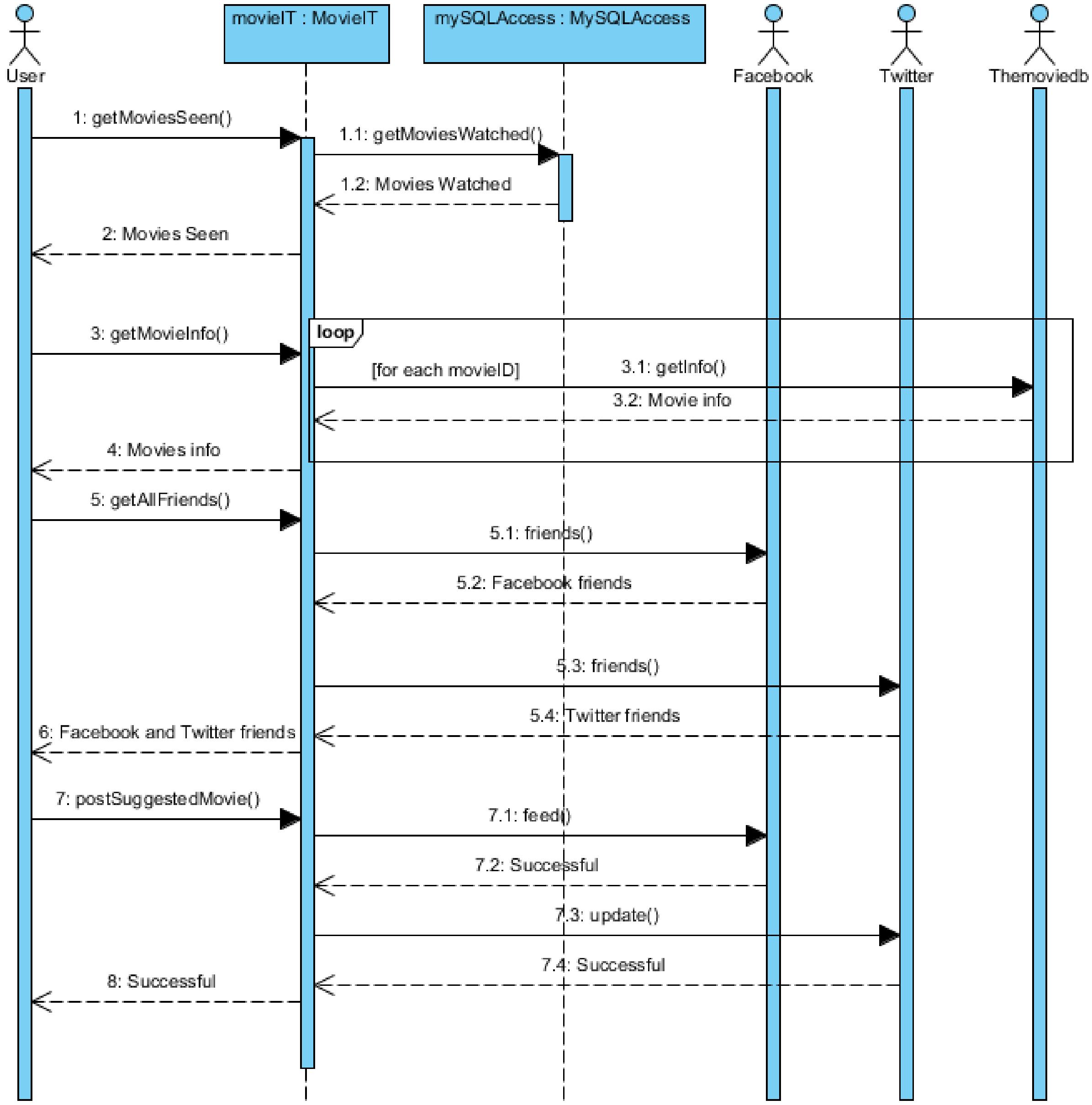


Fig 4.7 Sequence diagram

CHAPTER 5

PROJECT DESCRIPTION

5.1 APRIORI DATAFLOW

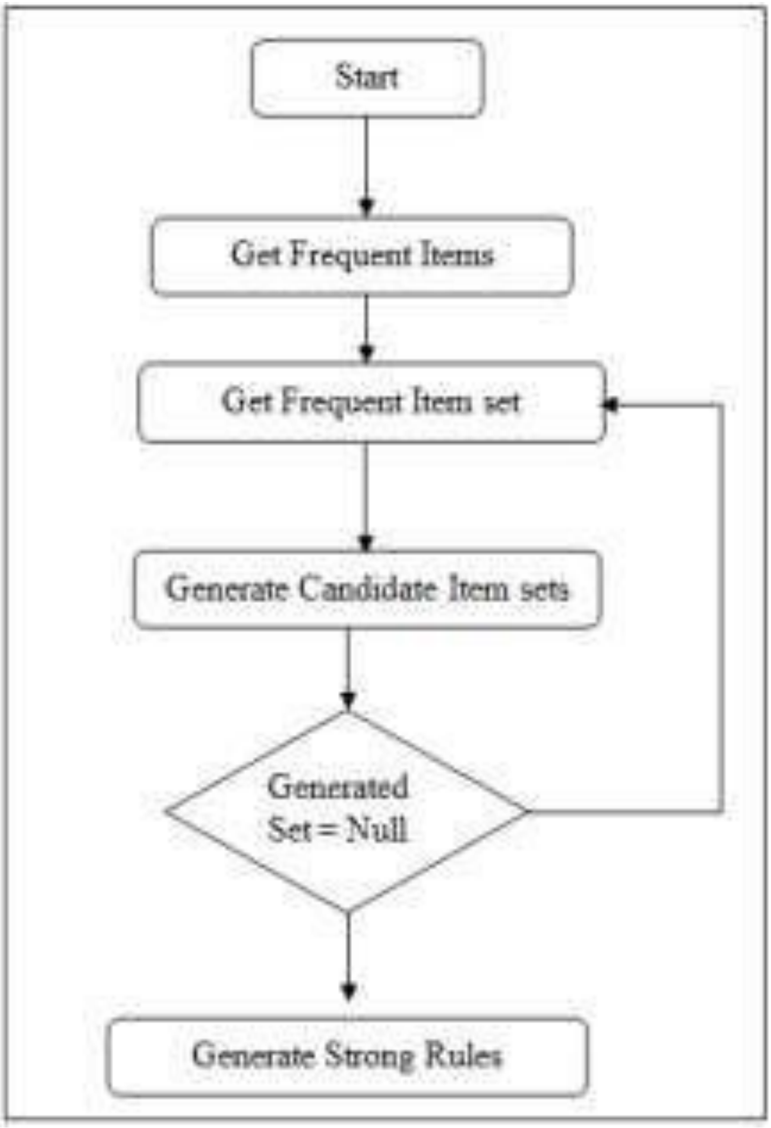


Fig 5.1 Apriori dataflow

5.2 MOVIE RECOMMENDATION ALGORITHM

5.2.1 APRIORI ALGORITHM:

In computer science and data mining, Apriori is a classic algorithm for learning association rules. Apriori is designed to operate on databases containing transactions. As is common in association rule mining, given a set of itemsets, the algorithm attempts to find subsets which are common to at least a minimum number C of the itemsets.

Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found.

Apriori uses breadth-first search and a tree structure to count candidate item sets efficiently. It generates candidate item sets of length k from item sets of length k − 1. This paper emphasis the future of data mining starting from the classic definition of “data mining” to the new trends in data mining.

The major reason behind data mining‟s great deal of attraction and attention in information industry in recent years, is due to the wide availability of huge amounts of data, and the eminent need for turning such data into useful information and knowledge. The information and knowledge gained can be used for applications ranging from business management, production control, and market analysis, to engineering design and science exploration.

Then it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent k-length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates.

The best example of Apriori algorithm is market basket analysis.

Market Basket Analysis is a modelling technique based upon the theory that if you buy a certain group of items, you are more (or less) likely to buy another group of items. For example, if you are in an English pub and you buy a pint of beer and don't buy a bar meal, you are more likely to buy crisps.

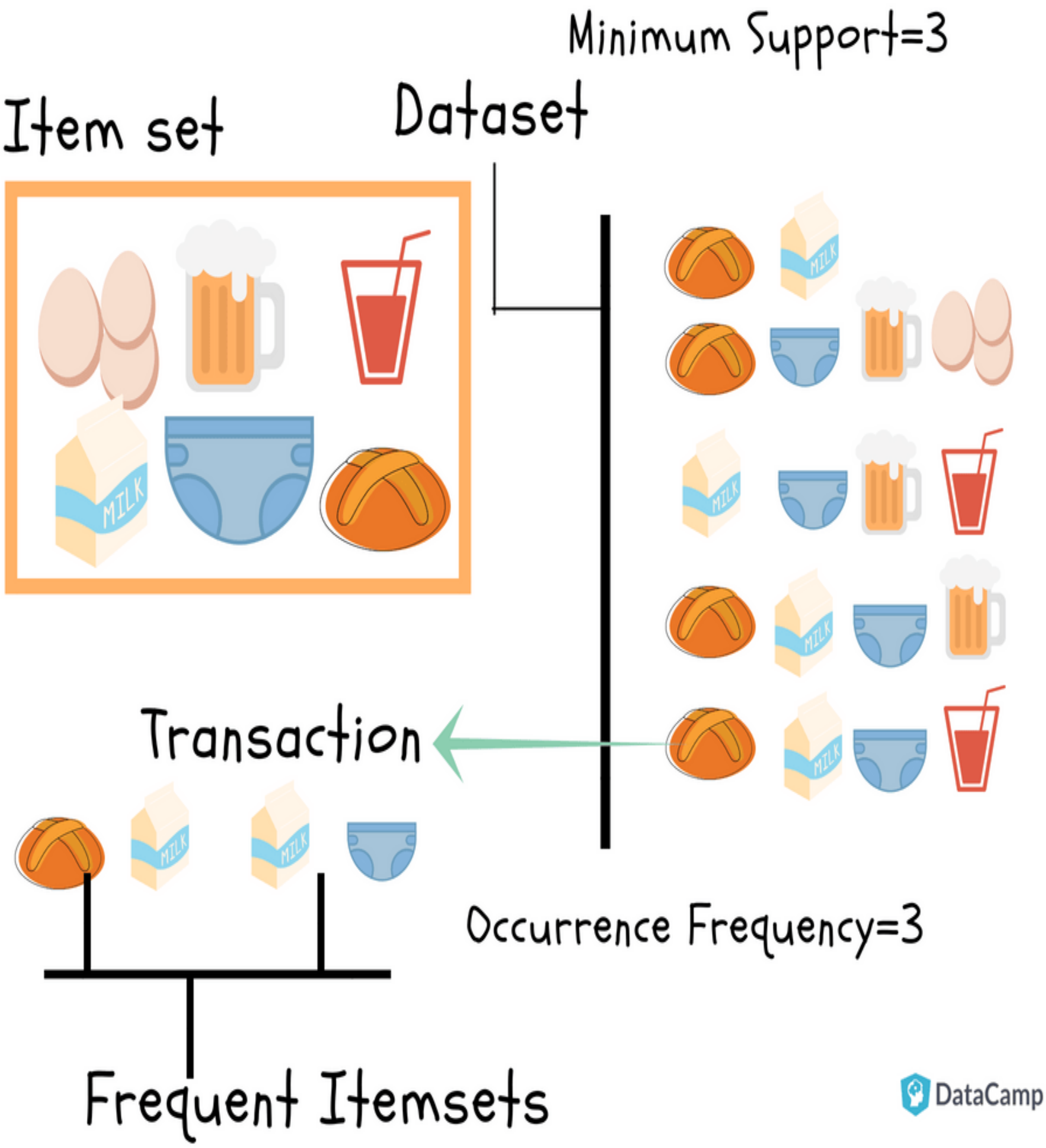


Fig 5.2.1 Market basket analysis

The Apriori Algorithm is an influential algorithm for mining frequent itemsets for boolean association rules. Key Features :

- Frequent Itemsets: The sets of item which has minimum support (denoted by L_i for i^{th}-Itemset). • Apriori Property: Any subset of frequent itemset must be frequent.
- Join Operation: To find L_k, a set of candidate k-itemsets is generated by joining L_{k-1} with itself.

Let X, $Y \subseteq I$ be any two itemsets. Observe that if $X \subseteq Y$, then $sup(X) \geq sup(Y)$, which leads to the following two corollaries:

- *If X is frequent, then any subset $Y \subseteq X$ is also frequent.*
- *If X is not frequent, then any superset $Y \supseteq X$ cannot be frequent.*

Based on the above observations, we can significantly improve the item-set mining algorithm by reducing the number of candidates we generate, by limiting the candidates to be only those that will potentially be frequent. First we can stop generating supersets of a candidate once we determine that it is infrequent, since no superset of an infrequent itemset can be frequent. Second, we can avoid any candidate that has an infrequent subset. These two observations can result in significant pruning of the search space.

- *Find frequent set L_{k-1}.*
- *Join Step.*
  - *C_k is generated by joining L_{k-1} with itself*
- *Prune Step.*
  - *Any $(k-1)$ -itemset that is not frequent cannot be a subset of a frequent k -itemset, hence should be removed.*

Where

- *(C_k: Candidate itemset of size k)*
- *(L_k: frequent itemset of size k)*

Apriori Pseudo code

C_k: Candidate itemsets of size K
L_k frequent itemsets of size K
L_1 = {frequent items};
For** (k = 1; L_k! = ∅; k++) **do
Begin
C_{k+1} = apriori_gen (L_k, min_sup);
***For each** transaction t in database **do** // scan D for counts*
Increment the count of all candidates in C_{k+1} that are contained in t
L_{k+1} = candidates in C_{k+1} with min_support
End
***Return** $\cup_k L_k$;*

Fig 5.2.1 Apriori pseudo code

5.3 PSEUDOCODE

```
import time
import operator
from itertools import islice, compress,ifilterfalse
from collections import namedtuple, defaultdict
from sets import Set
import itertools
all_genres = ['unknown', 'Action', 'Adventure', 'Animation', 'Children',
'Comedy', 'Crime', 'Documentary', 'Drama', 'Fantasy', 'Film-Noir',
'Horror', 'Musical', 'Mystery', 'Romance', 'Sci-Fi', 'Thriller',
'War', 'Western']
def _parse_movies(movie_file):
movie_attrs = ['id', 'name', 'release_date', 'unknown_field', 'imdb_url']
movies = {}
for line in movie_file:
tokens = line.split('|')
if not tokens or len(tokens) < 3:
continue
movie = dict(zip(movie_attrs, tokens))
genre_bools = map(int, tokens[len(movie_attrs) : len(tokens)])
genres = Set(compress(all_genres, genre_bools))
movie['genres'] = genres
movie['release_date'] = coerce_date(movie['release_date'])
movie['number_of_ratings'] = 0
movie['sum_of_ratings'] = 0
id = movie['id'] = int(movie['id'])
movies[id] = movie
return movies
def coerce_date(date_as_str):
```

```
date_as_str = date_as_str.strip()
if date_as_str:
return time.strptime(date_as_str, '%d-%b-%Y')
else:
return None

Rating = namedtuple('Rating', ['userid', 'movieid', 'rating', 'timestamp'])
def _ratings_iterator(ratings_file):
for line in ratings_file:
tokens = line.split()
if not tokens:
continue
tokens = map(long, tokens)
rating = Rating(*tokens)
yield rating
def get_movies(movie_file, ratings_file):

movies = _parse_movies(movie_file)
ratings = _ratings_iterator(ratings_file)
ratings_for_valid_movies = filter(lambda x : x.movieid in movies, ratings)
for rating in ratings_for_valid_movies:
movies[rating.movieid]['sum_of_ratings'] += rating.rating
movies[rating.movieid]['number_of_ratings'] += 1
return movies
def most_watched_movie(movies):
return max(movies, key=(lambda m : m['number_of_ratings']))
def most_watched_genre(movies):
by_genre = defaultdict(int)
for movie in movies:
for genre in movie['genres']:
```

```
by_genre[genre] += movie['number_of_ratings']
return max(by_genre.iterkeys(), key=(lambda genre : by_genre[genre]))
def most_popular_movie(movies):
return max(movies, key=(lambda m : m['sum_of_ratings']))
def most_popular_movie_for_genre(movies, genre):
movies_in_genre = itertools.ifilter(lambda m : genre in m['genres'],
movies)
return most_popular_movie(movies_in_genre)


def most_popular_movie_for_genre_and_year(movies, genre, year):
movies_with_release_date = itertools.ifilter(lambda m :
operator.truth(m['release_date']), movies)
movies_in_year = itertools.ifilter(lambda m : m['release_date'].tm_year
== year, movies_with_release_date)
movies_in_year_in_genre = itertools.ifilter(lambda m : genre in
m['genres'], movies_in_year)
# At this point, the list of movies is going to be very small, and possibly
empty
# We cannot send an empty generator to max() function call, it fails
horribly
# So, we convert the generator to a in-memory tuple, and then call
most_popular_movie
movies_in_year_in_genre = tuple(movies_in_year_in_genre)
if movies_in_year_in_genre:
return most_popular_movie(movies_in_year_in_genre)
else:
return None
def get_all_years(movies):
movies_with_release_date = itertools.ifilter(lambda m :
```

```
operator.truth(m['release_date']), movies)
years_possibly_duplicate = itertools.imap(lambda m :
m['release_date'].tm_year, movies_with_release_date)
return sorted(list(unique_everseen(years_possibly_duplicate)))
def unique_everseen(iterable, key=None):
"List unique elements, preserving order. Remember all elements ever
seen."
# unique_everseen('AAAABBBCCDAABBB') --> A B C D
# unique_everseen('ABBCcAD', str.lower) --> A B C D
seen = set()
seen_add = seen.add
if key is None:
for element in ifilterfalse(seen.__contains__, iterable):
seen_add(element)
yield element
else:
for element in iterable:
k = key(element)
if k not in seen:
seen_add(k)
yield element


def main():
movies = get_movies(open('movies/movie.data'),
open('movies/ratings.data'))
movies = movies.values()
print("Most Popular Movie : %s" %
most_popular_movie(movies)['name'])
print
```

```
print("Most Watched Movie : %s" %
most_watched_movie(movies)['name'])
print
print("Most Watched Genre : %s" % most_watched_genre(movies))
print
print("Popular Movie By Genre")
print("=======================")
for genre in all_genres:
print("%s : %s" % (genre, most_popular_movie_for_genre(movies,
genre)['name']))
print
for year in get_all_years(movies):
print
print("Year %s" % year)
print("=======================")
for genre in all_genres:
movie = most_popular_movie_for_genre_and_year(movies, genre, year)
if movie:
movie_name = movie['name']
else:
movie_name = "No Movie"
print("%s : %s" % (genre, movie_name))
if __name__ == '__main__':
main()
```

//code for recommendation and graphs

```
import
unittest
        from movies import coerce_date, _ratings_iterator, _parse_movies
        class MoviesTests(unittest.TestCase):
        def test_coerce_date(self):
        t = coerce_date(' 01-Nov-1995 ')
        self.assertEqual(t.tm_year, 1995)
        self.assertEqual(t.tm_mon, 11)
        self.assertEqual(t.tm_mday, 1)
        t = coerce_date('')
        self.assertIsNone(t)
        def test_parse_movies(self):
        movies_file = """
        1|Toy Story (1995)|01-Jan-1995||http://us.imdb.com/M/title-
        exact?Toy%20Story%20(1995)|0|0|0|1|1|1|0|0|0|0|0|0|0|0|0|0|0|0|0
        14|Movie Without Release Date| ||http://us.imdb.com/M/title-
        exact?Postino,%20Il%20(1994)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|1|0|0|0|0
        """
        movies = _parse_movies(StringIO(movies_file))
        self.assertTrue(1 in movies)
        self.assertTrue(14 in movies)
        toystory = movies[1]
        no_release_date = movies[14]
```

```
self.assertEquals(toystory['name'], 'Toy Story (1995)')
self.assertEquals(toystory['release_date'].tm_year, 1995)
self.assertEquals(toystory['imdb_url'], 'http://us.imdb.com/M/title-exact?Toy%20Story%20(1995)')
toystory_genres = ['Animation', 'Children', 'Comedy']
for g in toystory_genres:
self.assertTrue(g in toystory['genres'])
self.assertEquals('Movie Without Release Date', no_release_date['name'])
self.assertIsNone(no_release_date['release_date'])
def test_ratings_iterator(self):
ratings_file = """
196 242 3 881250949
186 302 3 891717742
22 377 1 878887116
244 51 2 880606923
166 346 1 886397596
"""
ratings = tuple(_ratings_iterator(StringIO(ratings_file)))
self.assertEqual(ratings[1].userid, 186)
self.assertEqual(ratings[1].movieid, 302)
self.assertEqual(ratings[1].rating, 3)
self.assertEqual(ratings[1].timestamp, 891717742)
```

```
self.assertEqual(ratings[3].userid, 244)

self.assertEqual(ratings[3].movieid, 51)

self.assertEqual(ratings[3].rating, 2)

self.assertEqual(ratings[3].timestamp, 880606923)

if __name__ == '__main__':

unittest.main()
```

5.3 UI SNAPSHOTS

Menu Options: -

1.New User, 2. Existing_User, 3.User Details, 4. GENRE SELECTION GRAPH, 5.Exit =

Fig 5.3[a] Menu Options

Option 1: -

```
1.New User, 2. Existing_User, 3.User Details, 4. GENRE SELECTION GRAPH, 5.Exit = 1
1. Action
2. Adventure
3. Fantasy
4. Sci-Fi
5. Thriller
6. Romance
7. Animation
8. Comedy
9. Family
10. Musical
11. Mystery
12. Western
13. Drama
14. History
15. Sport
16. Crime
17. Horror
18. War
19. Biography
20. Music
21. Documentary
22. Game-Show
23. Reality-TV
24. News
25. Short
26. Film-Noir
27. Romance
28. Action
29. Drama
30. Musical
Please, Choose a Genre = News
Congrats, User Created
```

Fig 5.2[b] Option 1

Option 2:-

```
1.New User, 2. Existing_User, 3.User Details, 4. GENRE SELECTION GRAPH, 5.Exit = 2
Enter the User Number, There are 1 present = 0

1. Capitalism: A Love Story
2. The Square
3. Food Chains
1. Continue to Same User, 2. Go back ? = 1

Pick a movie from above List= The Square
```

Fig 5.2[c] Option 2[a]

```
Suggested Movies, 1. The Square
Suggested Movies, 2. Tangled
Suggested Movies, 3. Frozen
Suggested Movies, 4. Happy Feet 2
Suggested Movies, 5. Rio 2
Suggested Movies, 6. Home on the Range
Suggested Movies, 7. The Princess and the Frog
Suggested Movies, 8. The Hunchback of Notre Dame
Suggested Movies, 9. Mulan
```

Fig 5.2[c] Option 2[b]

```
Suggested Movies, 1713. Welcome to Collinwood
Suggested Movies, 1714. Snatch
Suggested Movies, 1715. My Cousin Vinny
Suggested Movies, 1716. Let's Go to Prison
Suggested Movies, 1717. Dum
Available Genre :- ['Documentary', 'Drama', 'History', 'News']
Suggested Genre :- ['News', 'Music', 'Sci-Fi', 'Action', 'Mystery', 'Romance', 'History', 'Adventure', 'Biography', 'Fantasy', 'War',
 'Family', 'Documentary', 'Musical', 'Animation', 'Drama', 'Thriller', 'Horror', 'Crime']
Above Listed movies has rating is more or equals to 4.9

1. Continue to Same User, 2. Go back ? =
```

Fig 5.2[c] Option 2[c]

```
Suggested Movies, 1715. My Cousin Vinny
Suggested Movies, 1716. Let's Go to Prison
Suggested Movies, 1717. Dum
Available Genre :- ['Documentary', 'Drama', 'History', 'News']
Suggested Genre :- ['News', 'Music', 'Sci-Fi', 'Action', 'Mystery', 'Romance', 'History', 'Adventure', 'Biography', 'Fantasy', 'War',
 'Family', 'Documentary', 'Musical', 'Animation', 'Drama', 'Thriller', 'Horror', 'Crime']
Above Listed movies has rating is more or equals to 4.9

1. Continue to Same User, 2. Go back ? = 1

Pick a movie from above List= Project X

Suggested Movies, 1. Tangled
Suggested Movies, 2. Monsters University
Suggested Movies, 3. Toy Story 3
Suggested Movies, 4. The Good Dinosaur
Suggested Movies, 5. Brave
Suggested Movies, 6. WALL·E
Suggested Movies, 7. A Christmas Carol
Suggested Movies, 8. Up
```

Fig 5.2[c] Option 2[d]

```
Suggested Movies, 660. The White Ribbon
Suggested Movies, 661. The Dead Zone
Suggested Movies, 662. Ondine
Available Genre :- ['Comedy', 'Crime']
Suggested Genre :- ['Animation', 'Biography', 'Comedy', 'Action', 'Thriller', 'Family', 'Crime', 'Mystery']
Above Listed movies has rating is more or equals to 6.7

1. Continue to Same User, 2. Go back ? = 1
```

Fig 5.2[c] Option 2[e]

Option 3: -

```
1. Continue to Same User, 2. Go back ? = 2
1.New User, 2. Existing_User, 3.User Details, 4. GENRE SELECTION GRAPH, 5.Exit = 3
Enter the User Number, There are 1 present = 0
Till now user has watched  Following Movies :-

1. The Square
2. Project X
3. K-PAX
Till now user has selected Following Genres :-

1. ['Documentary', 'Drama', 'History', 'News']
2. ['Comedy', 'Crime']
3. ['Drama', 'Mystery', 'Sci-Fi']
```

Fig 5.2[d] Option 3[a]

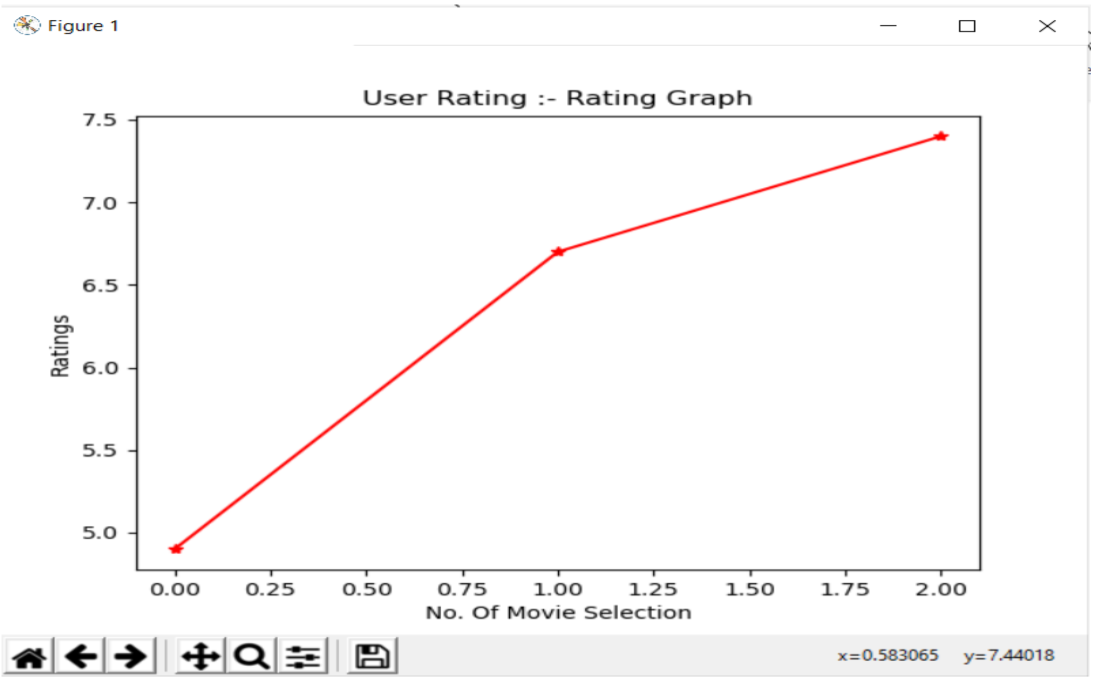


Fig 5.2[d] Result analysis

Option 4: -

```
1.New User, 2. Existing_User, 3.User Details, 4. GENRE SELECTION GRAPH, 5.Exit = 4
User 1.

  1. ['Documentary', 'Drama', 'History', 'News']
  2. ['Comedy', 'Crime']
  3. ['Drama', 'Mystery', 'Sci-Fi']
```

Fig 5.2[e] Option 4

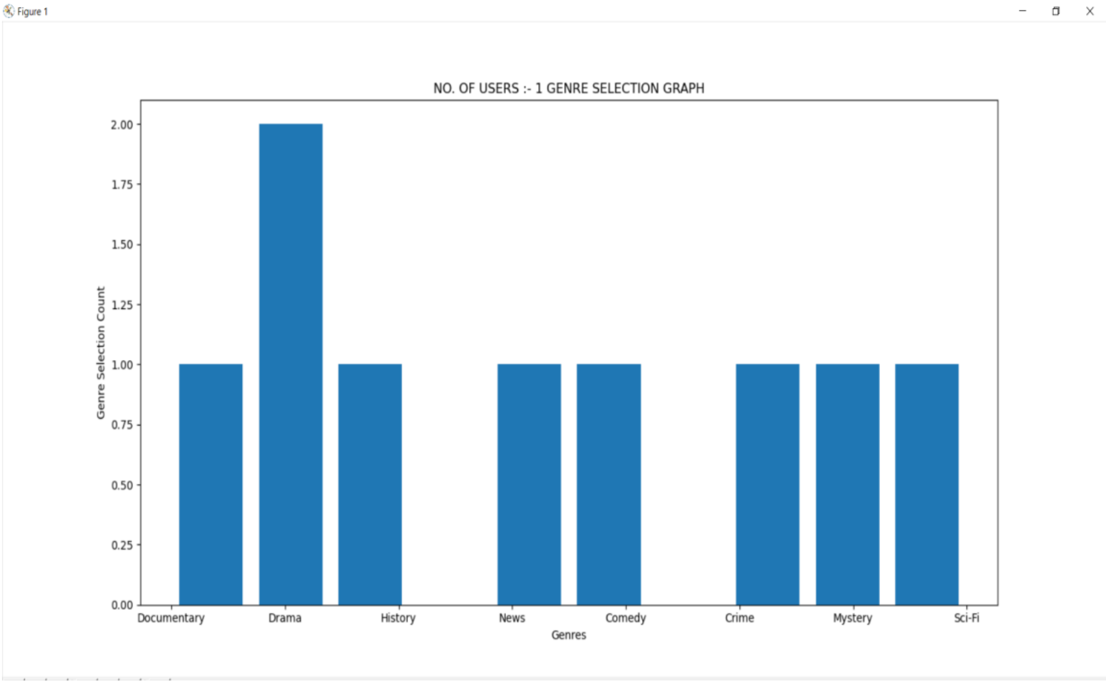


Fig 5.2[e] Result analysis

Option 5: -

```
1.New User, 2. Existing_User, 3.User Details, 4. GENRE SELECTION GRAPH, 5.Exit = 5


Process finished with exit code 0
```

Fig 5.2[e] Result analysis

CHAPTER 6

CONCLUSION

The phase 1 was a learning phase in which we as students got to know a lot of things. This was a very nice exposure to learn a lot of new concepts. Phase 1 was more like a learning experience for the domain of data mining, statistics and machine learning. The main aspect of the major project was seen in this phase, a lot of learning and understanding has gone into doing this.

The phase 2 is a closer approach to the project where the actual implementation takes place. The Movie recommendation system is a major way of assessment in the current scenario. The previously envisioned functionalities have been partially implemented and can be used. The software has a desktop application in which the users can add details like name, user_id. This software provides a search engine which makes the search process easier.

It is beyond doubt that the further development potential of this software system is great and by seizing this opportunity, when it will be completed and released, it could play a considerable role in the future.

REFERENCES

[1] [1] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,” *IEEE Trans. Knowl. Data Eng.*

[2] G. Linden, B. Smith, and J. York, “Amazon recommendations: Itemto-item collaborative filtering,” *IEEE Internet Comput.*, Feb. 2003.

[3] Michael Hashler, “Recommender Lab: A Framework for Developing and Testing Recommendation Algorithms” *Nov. 2011.*

[4] R. Bell, Y. Koren, and C. Volinsky, “Modeling relationships at multiple scales to improve accuracy of large recommender systems” KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, 2007, ACM.

[5] O. Celma and P. Herrera, “A new approach to evaluating novel recommendations”, RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems, New York, NY, USA, 2008, ACM.

[6] C. N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, “Improving recommendation lists through topic diversification”, WWW '05: Proceedings of the 14th international conference on World Wide Web, New York, NY, USA, 2005, ACM.

Movie Recommendation using Machine Learning

by Sadiya Saba

Submission date: 15-May-2020 12:06PM (UTC+0530)
Submission ID: 1324798308
File name: 1NH16CS749.pdf (608.27K)
Word count: 6642
Character count: 37331

PROJECT TITLE: "MOVIE RECOMMENDATION USING MACHINE LEARNING"

TEAM MEMBERS:

Sadiya Saba(1NH16CS749)

Rashmi K(1NH16CS748)

Anusha S Bachihal(1NH16CS738)

Semester and Section:8C

CHAPTER 1

INTRODUCTION

As we realize that, the world is becoming quicker more than ever. Everybody is scrambling for their definitive objectives. This thirst results into the improvement of pretty much every area. Online business is one of them. We individuals, don't have the opportunity to shop from market and this isn't the end. We don't have the opportunity to pick the article from the assortment. This made the incipient organism of web based shopping, which these days, turned into an enormous tree, of huge amounts of branches.

As the online market develops exponentially, clearly rivalry will entered in different fields moreover. Presently, proprietors of their individual destinations need to draw in their clients by giving alluring offices. Recommender Engines is one of the offices given to clients. Recommender motor are the most promptly conspicuous AI procedure being used today. We will have seen administrations or destinations that endeavor to suggest books or motion pictures or articles dependent on our past activities. They attempt to induce tastes and inclinations and recognize obscure things that are of intrigue.

Because of advances in the recommender framework clients continually anticipate great proposals. They have a low limit for administrations that can't make proper recommendations. In the event that a music spilling application can't foresee and play music that the client likes, at that point the client will basically quit utilizing it. This has prompted a high accentuation by tech organizations on improving their suggestion frameworks. Be that as it may, the issue is surprisingly unpredictable. Each client has various inclinations like what's more even the flavor of a solitary client can change contingent upon enormous number of elements, for example, positive state of mind, season, or kind of action the client is doing. Since every client is unique, this methodology is viewed as excessively basic.

/The fundamental thought behind this framework is that motion pictures that are progressively famous and widely praised will have a higher likelihood of being preferred by the normal audience.For model, the sort of music one might want to hear while practicing varies incredibly from the kind of music he'd tune in to when preparing supper. Netflix comparably suggests DVDs that might be of intrigue, and broadly prize to scientists who could improve the nature of their proposals. Long range informal communication locales like Facebook use variations on recommender.

This paper depends on proposal framework that prescribes various things to clients. This framework will prescribe motion pictures to clients. This framework will give progressively exact outcomes when contrasted with the current frameworks. The current framework deals with singular clients' appraising. This might be at some point futile for the clients who have diverse taste from the proposals appeared by the framework as each client may have various tastes. This framework figures the likenesses between various clients and afterward prescribe film to them according to the evaluations given by the various clients of comparative tastes. This will give an exact suggestion to the client.

This is an online just as android framework where there is a film web administration which offers types of assistance to client to rate motion pictures, see proposals put remarks and see comparative motion pictures. There are frameworks which manage the self-suggestion as opposed to thinking about the preferences of clients, we subsequently construct a framework that admissions the clients wishes and afterward suggests a watch-rundown of motion pictures which depends on their chose class. What's more, hence makes the watch increasingly best and pleasant to the client.

/Given a lot of clients with their past appraisals for a lot of motion pictures, would we be able to foresee the rating they will allocate to a film they have not recently evaluated?

Ex. "Which film will you like" given that you have seen 'Harry Potter and the Sorcerer's Stone', 'Harry Potter and the Chamber of Secrets', 'Harry Potter and the Prisoner of Azkaban' and clients who saw these motion pictures likewise loved "Harry Potter and the Goblet of Fire"?

Recommender frameworks are data separating devices that seek to anticipate the rating for clients and things, overwhelmingly from huge information to suggest their preferences. Film proposal frameworks give a system to help clients in arranging clients with comparable interests. The reason for a suggestion framework fundamentally is to look for content that would be fascinating to a person. Besides, it includes various variables to make customized arrangements of valuable and intriguing substance explicit to every client/person.

Suggestion frameworks are Artificial Intelligence based calculations that skim through every conceivable choice and make an altered rundown of things that are intriguing and applicable to a person. These outcomes depend on their profile, search/perusing history, what others with comparable qualities/socioeconomics are viewing, and how likely are you to watch those motion pictures. This is accomplished through prescient demonstrating and heuristics with the information accessible.

- Improve maintenance

Takes into account the client's inclinations and keeps them snared to the application.

- Increase deals

Can improve business by an incredible edge by giving different suggestions of various things.

Form propensities

Affecting use design in clients.

Accelerate work

Helps the examiners for additional exploration and decreases their work.

The fundamental thought behind this framework is that films that are increasingly mainstream and widely praised will have a higher likelihood of being loved by the normal crowd. Second is content-based sifting, where we attempt to profile the client's advantages utilizing data gathered, and suggest things dependent on that profile.

- Create a client account.
- Record his/her history.
- Based on the history suggest more motion pictures.
- Based on his past rating, suggest motion pictures.
- Also suggests motion pictures dependent on comparative type.
- Can track the favored IMDB appraised films dependent on his history.

- Can track the most favored film sort among n clients.

A. Administrator

The framework administrator will include film in a database, see motion pictures and update it.

B. Proposal Engine

This proposal motor will compute the likenesses between the various clients. Based on that similitudes determined, this motor will prescribe film to a client.

C. Film Web Service

This will permit client to rate motion pictures, remarks on motion pictures. This administration will likewise show the

film suggestion to the clients.

D. Client

The android client can rate a film, can remark on any film, and can see comparable motion pictures suggested by different clients who are like this client.

CHAPTER 2

LITERATURE SURVEY

Over the previous decade, an enormous number of suggestion frameworks for an assortment of spaces have been created and are being used. These proposal frameworks utilize an assortment of strategies, for example, content based methodology, shared methodology, information based methodology, utility based methodology, half breed approach, and so on. The majority of the online suggestion frameworks for an assortment of things use evaluations from past clients to make proposals to current clients with comparable interests. One such framework was planned by Jung, Harris, Webster and Herlocker (2004) for improving indexed lists. The framework urges clients to enter longer and increasingly useful hunt inquiries, and gathers appraisals from clients with regards to whether indexed lists meet their data need or not. These appraisals are then used to make suggestions to later clients with comparable necessities.

Current Social Networking World Internet person to person communication destinations, which started in 1995 with Classmates.com, have flooded in notoriety and use through verbal publicizing. From that point forward, a wide scope of virtual networks have shaped filling various needs and focusing on differing specialty crowds:

Specifically, we've decided to investigate the film specialty as this is a region where our venture can huge enhancements contrasted with existing items and frameworks. Customary film sites (IMDB, AOL Movies) work by demonstrating worldwide client evaluations on motion pictures in their database. Motion pictures are classified by metadata, for example, sort, period, executives, etc. Clients can scan for films, peruse records and read surveys composed by pundits or different clients. Be that as it may, the vast majority of these administrations come up short on any close to home proposal framework and haven't exploited long range interpersonal communication networks or group astuteness. A few sites, for example, Blockbuster, do give individualized suggestions dependent on a client's appraisals yet donot incorporate any informal communication part. Yippee! Motion pictures goes further and utilizes individual evaluations to recommend motion pictures as of now playing in theater, on TV, and out on DVD. It additionally draws upon its huge client base to give arrangements of comparable film fans, their appraisals, and surveys. Other film destinations, as Flixster, adopt an alternate strategy. Flixster structures electronic networks around films and proposes motion pictures to watch dependent on what your companions have appraised.

The framework is based on windows 2007 working framework. The framework utilizes propelled java innovation alongside AI ideas. MySQL is utilized for putting away information. This framework utilizes three-level engineering. The web administration layer gives the android client to rate motion pictures, see comparative proposals given by the framework and remark on it.

The proposed framework is a superior framework than some other existing frameworks. This framework has included the positive highlights of existing frameworks and has conquered the downsides of existing frameworks. The present framework can:

- Create a client account.
- Record his/her history.
- Based on the history suggest more motion pictures.

- Based on his past rating, suggest motion pictures.

- Also suggests motion pictures dependent on comparative class.

- Can track the favored IMDB appraised motion pictures dependent on his history.

- Can track the most favored film type among n clients.

The framework utilizes all the current calculations for example content based, setting based and community oriented based calculations. Every one of these calculations are consolidated to give increasingly exact outcome. We here utilize the Apriori calculation to make this conceivable. Apriori calculation is utilized for finding continuous thing sets in a dataset for Boolean affiliation rule. Name of the calculation is Apriori on the grounds that it utilizes earlier information on visit thing sets properties. Apriori calculation is applied when there are a few things of connections that need to be broke down.

Testing information mining handling then from the consequences of item yield proposals proper for clients will be done bit by bit by taking a gander at the following client buy history information in a specific period. Apriori property which helps by lessening the pursuit space.

The accompanying modules are created as:

The framework administrator will include film in a database, see motion pictures and update it. Keeps the tab of the client history as well.

This proposal motor will figure the likenesses between the various clients. Based on that similitudes determined, this motor will prescribe film to a client.

In view of the chose classification motion pictures are suggested and the chose kind is put away for additional proposals.

This will permit client to rate motion pictures, remarks on motion pictures. This administration will likewise show the film suggestion to the clients.

The android client can rate a film, can remark on any film, and can see comparable motion pictures suggested by different clients who are like this client. Can see the chart on his watch history and locate the most favored sort.

The achievability of the venture is broke down in this stage and strategic agreement is advanced with a general arrangement for the undertaking and some quotes. During framework examination the plausibility investigation of the proposed framework is to be completed. This is to guarantee that the proposed framework isn't a weight to the organization. For achievability investigation, some comprehension of the significant prerequisites for the framework is basic.

Three key contemplations associated with the plausibility examination are,

- ECONOMICAL FEASIBILITY

- TECHNICAL FEASIBILITY

- SOCIAL FEASIBILITY

This investigation is completed to check the financial effect that the framework will have on the association. The measure of store that the organization can fill the innovative work of the framework is restricted. The consumptions must be advocated. Along these lines the created framework also inside the financial plan and this was accomplished on the grounds that the majority of the innovations utilized are uninhibitedly accessible. Just the tweaked items must be bought.

This examination is done to check the specialized possibility, that is, the specialized necessities of the framework. Any framework created must not have a popularity on the accessible specialized assets. This will prompt levels of popularity on the accessible specialized assets. This will prompt levels of popularity being set on the customer. The created framework must have an unassuming necessity, as just negligible or invalid changes are required for actualizing this framework.

The part of study is to check the degree of acknowledgment of the framework by the client. This incorporates the way toward preparing the client to utilize the framework proficiently. The client must not feel undermined by the framework, rather should acknowledge it as a need. The degree of acknowledgment by the clients exclusively relies upon the strategies that are utilized to instruct the client about the framework and to make him acquainted with it. His degree of certainty must be raised with the goal that he is likewise ready to make some valuable analysis, which is invited, as he is the last client of the framework.

CHAPTER 3

REQUIREMENT ANALYSIS

The customer application is the connection between the client and the server application. Its errand is to assemble data from the clients and to permit clients to play motion pictures. The data is sent to the server application, where it is put away, and later used to create suggestions.

Moreover, the data is utilized to gauge recommender exactness. This considers examination of how accuracy is affected by various recommender techniques.

The prerequisites for the customer application are:

The customer application will give an interface that makes it conceivable to play motion pictures by choice, or by route through standard film player catches like play, delay, stop and skip.

The customer application will make it conceivable to demand proposals and to send the solicitations to the server application.

The customer application will make it conceivable to assess every film and to send this data to the server application. Can likewise rate it, see it, subsequently all the client movement is put away as such for additional references.

Application server structures are programming systems for building application servers. An application server system gives the two offices to make web applications and a server domain to run them. The server application gets data from the customer application, and furnishes the customer application with proposals.

The prerequisites for the server application are:

The server application will get and deal with demands for proposals.

The server application will get and store film assessments.

:

The server application will be equipped for delivering proposals by deciphering logical data given by the clients, and assessments given by the real client and other comparative clients. Gives out the successive films chose. We here store the produced yield in the class objects for future concerns. This calculation is for the thing set based filtration.

Non-utilitarian necessity is a prerequisite that indicates standards that can be utilized to pass judgment on the activity of a framework, as opposed to explicit practices. They are stood out from practical necessities that characterize explicit conduct or capacities. The arrangement for actualizing utilitarian prerequisites is nitty gritty in the framework structure.

Will deliver exact proposals that coordinate the client's film inclination.

The customer application will limit nosiness and simultaneously catch client consideration with the goal that a worthy measure of assessment information is gotten.

The recommender framework will have the capability of being adaptable both regarding size and topography

Some Non-Functional Requirements are as per the following:

- Reliability
- Maintainability
- Performance
- Portability
- Scalability
- Flexibility

Openness is a general term used to depict how much an item, device, administration, or condition is available by whatever number individuals as could reasonably be expected.

In my undertaking, understudies and resources can login from any framework as this is a desktop application, it isn't framework subordinate. UI is straightforward and efficient and simple to utilize.

In programming designing, practicality is the straightforwardness with which a product product can be adjusted so as to:

- Correct surrenders

- Meet new necessities

New functionalities can be included in the undertaking based the client prerequisites.

Since the writing computer programs is basic, it is simpler to discover and address the deformities and to roll out the improvements in the venture.

Framework is fit for dealing with increment all out throughput under an expanded burden when assets (regularly equipment) are included. Framework can work regularly under situations, for example, low data transfer capacity and huge number of clients.

Transportability is one of the key ideas of elevated level programming. Convenientce is the software code base component to have the option to reuse the current code as opposed to creating new code while moving programming from a situation to another. Undertaking can be executed under various activity conditions gave it meets its base arrangements. Just framework documents and dependant gatherings would have to be arranged in such case.

Programming Reliability is the likelihood of disappointment free programming activity for a specified timeframe in a predetermined domain. Programming Reliability is likewise a important factor influencing framework unwavering quality.

Python is a broadly useful deciphered, intuitive, object-situated, and elevated level programming language. A deciphered language, Python has a structure reasoning that accentuates code lucidness (strikingly utilizing whitespace space to delimit code squares as opposed to wavy sections or catchphrases), and a grammar that permits developers to communicate ideas in less lines of code than may be utilized in dialects, for example, C++or Java. It gives builds that empower clear programming on both little and huge scopes. Python translators are accessible for some working frameworks. CPython, the reference usage of Python, is open source programming and has a network based advancement model, as do about the entirety of its variation executions.

Python is overseen by the non-benefit Python Software Foundation. Python includes a powerful sort framework and programmed memory the board. It bolsters different programming ideal models, including object-arranged, basic, useful and procedural, and has an enormous and far reaching standard library.

An informational index (or dataset) is an assortment of information. On account of even information, an informational index compares to at least one database tables, where each segment of a table speaks to a specific variable, and each line relates to a given record of the informational index being referred to.

For the venture we have used the film dataset from 'Film Lens(Kaggle)'.

• Movie Lens is an informational collection that gives 10000054 client appraisals on films.

• 95580 labels applied to 10681 motion pictures by 71567 clients.

• Users of Movie Lens were chosen arbitrarily.

• All clients appraised in any event 20 films.

• Each client spoke to by a one of a kind id.

• u.info : The quantity of clients, things, and appraisals in the u information ssset.

• u.item : Information about the things

film id | film title | discharge date |

• u.genre : A rundown of the Action

Experience | Animation | Children's | Comedy | Crime | Documentary | Drama | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western type types.

u.user : Information about the clients

client id | age | orientation | occupation | postal division

```
1::Toy Story (1995)::Animation|Children's|Comedy
2::Jumanji (1995)::Adventure|Children's|Fantasy
3::Grumpier Old Men (1995)::Comedy|Romance
4::Waiting to Exhale (1995)::Comedy|Drama
5::Father of the Bride Part II (1995)::Comedy
6::Heat (1995)::Action|Crime|Thriller
7::Sabrina (1995)::Comedy|Romance
8::Tom and Huck (1995)::Adventure|Children's
9::Sudden Death (1995)::Action
10::GoldenEye (1995)::Action|Adventure|Thriller
11::American President, The (1995)::Comedy|Drama|Romance
12::Dracula: Dead and Loving It (1995)::Comedy|Horror
13::Balto (1995)::Animation|Children's
14::Nixon (1995)::Drama
15::Cutthroat Island (1995)::Action|Adventure|Romance
16::Casino (1995)::Drama|Thriller
17::Sense and Sensibility (1995)::Drama|Romance
18::Four Rooms (1995)::Thriller
19::Ace Ventura: When Nature Calls (1995)::Comedy
20::Money Train (1995)::Action
21::Get Shorty (1995)::Action|Comedy|Drama
22::Copycat (1995)::Crime|Drama|Thriller
23::Assassins (1995)::Thriller
24::Powder (1995)::Drama|Sci-Fi
25::Leaving Las Vegas (1995)::Drama|Romance
26::Othello (1995)::Drama
27::Now and Then (1995)::Drama
28::Persuasion (1995)::Romance
29::City of Lost Children, The (1995)::Adventure|Sci-Fi
```

In

the displaying some portion of the model, the dataset can be utilized this s known as multi-class grouping issue. Four highlights are remembered for the informational collection film name, IMDB rating, language, nation, year of discharge.

To comprehend model execution, it is a decent technique to isolate the informational index into preparing and test sets.Let's part the informational index utilizing the capacity train_test_split(). Capacities, target and test_set size of 3 parameters required.

CHAPTER 4

DESIGN

Framework engineering is the applied model that characterizes the structure, conduct, and more perspectives on a framework. An engineering portrayal is a conventional depiction and portrayal of a framework, sorted out such that supports thinking about the structures and practices of the framework.

- When a solicitation is made through the client through the application, the solicitation subsequently the solicitation is then acknowledged by the server.

- Server goes about as a medium between the client's advantage and his solicitation.

- The server at that point acknowledges it through the ip and the client is conceded their suggestions.

- Use case outlines model the usefulness of a framework utilizing on-screen characters and use cases. Use cases are a lot of activities, administrations, and capacities that the framework needs to perform.

- First work is to give input for example set of motion pictures with their subtleties like film name, language and to which type it has a place.

- Apriori is a calculation for visit thing set mining and affiliation rule learning over social databases. It continues by recognizing the incessant individual things in the database and extending them to bigger and bigger thing sets as long as those thing sets show up adequately frequently in the database.

- Then suggests the film dependent on Content-based Recommendation and

Community oriented Filtering Algorithm.

- Based on the recommendation given by the framework client will pick the film.

- A state chart is a graph utilized in software engineering to portray the conduct of a framework considering all the potential conditions of an item when an occasion happens. State outlines graphically speak to limited state machines. They are just used to comprehend object conduct all through the entire framework.

- In the above chart it portrays the conduct of film suggestion framework.

- When the information of film is given it checks for the comparable class by utilizing proposal calculation and recommends the film to client.

- Suggestion depends on the film evaluations, class, language and much of the time watched motion pictures.

- If the recently watched film's evaluating is three, at that point the recommended film's appraising ought to be equivalent or more noteworthy than three.

- Class outline in the Unified Modeling Language (UML) is a sort of static structure graph that depicts the structure of a framework by demonstrating the framework's classes, their characteristics, tasks (or strategies), and the connections among objects.

- The above class graph portrays the classes, characteristics and strategies for film proposal framework.

- Input film information: It is class which takes the information of the film subtleties.

- Apriori calculation: This class will do investigations dependent on the given information and recommends film.

- Movie choice: This class helps in choosing film dependent on the class and the film appraisals.

- Movie proposal: This class recommends the film to client dependent on the much of the time viewed movie(genre).

- Activity outline is characterized as an UML chart that centers around the execution and stream of the conduct of a framework rather than usage. It is likewise called object-arranged flowchart. Action charts comprise of exercises that are comprised of activities which apply to social demonstrating innovation.

- Firstly, need to enter the client qualifications.

- If another client needs to enlist.

- Then select the appropriate classification and the motion pictures are suggested.

- For this the framework utilizes the Apriori calculation, and subsequently the procedure proceeds and constantly another suggestion list is produced.

- A grouping outline basically portrays cooperation between objects in a successive request for example the request where these connections happen. We can likewise utilize the terms occasion graphs or occasion situations to allude to a grouping outline. Arrangement graphs depict how and in what request the articles in a framework work.

- Once a film is chosen by the client the server gets the subtleties of his recently watched motion pictures.

- The data is put away and in the event that we need to see the data again we can simply check for the client history utilizing the userid.

- For further the proposal can be imparted to companions on various stages.

- The age of suggestions goes on in circle until the client chooses one for his watch.

CHAPTER 5

PROJECT DESCRIPTION

In software engineering and information mining, Apriori is a great calculation for learning affiliation rules. Apriori is intended to work on databases containing exchanges. As is regular in affiliation rule mining, given a lot of itemsets, the calculation endeavors to discover subsets which are normal to in any event a base number C of the itemsets.

Apriori utilizes a "base up" approach, where visit subsets are expanded each thing in turn (a stage known as competitor age), and gatherings of applicants are tried against the information. The calculation ends when no further effective augmentations are found.

Apriori utilizes expansiveness first inquiry and a tree structure to tally up-and-comer thing sets effectively. It produces applicant thing sets of length k from thing sets of length k – 1. This paper accentuation the eventual fate of information mining beginning from the exemplary meaning of "information mining" to the new patterns in information mining.

The significant purpose for information mining‟s extraordinary arrangement of fascination and consideration in data industry as of late, is because of the wide accessibility of enormous measures of information, and the famous requirement for transforming such information into valuable data and information. The data and information picked up can be utilized for applications extending from business the executives, creation control, and market investigation, to building structure and science investigation.

At that point it prunes the applicants which have a rare sub design. As indicated by the descending conclusion lemma, the up-and-comer set contains all incessant k-length thing sets. From that point onward, it filters the exchange database to decide visit thing sets among the applicants.

The best case of Apriori calculation is advertise bushel examination.

Market Basket Analysis is a demonstrating strategy dependent on the hypothesis that in the event that you purchase a specific gathering of things, you are more (or more outlandish) to purchase another gathering of things. For instance, in the event that you are in an English bar and you purchase a 16 ounces of brew and don't purchase a bar feast, you are bound to purchase crisps.

The Apriori Algorithm is a persuasive calculation for digging successive itemsets for boolean affiliation rules. Key Features :

• Frequent Itemsets: The arrangements of thing which has least help (meant by Lifor ith-Itemset). • Apriori Property: Any subset of regular itemset must be visit.

• Join Operation: To discover Lk, a lot of competitor k-itemsets is created by joining Lk-1 with itself.

```
Imp0rt tlme
Imp0rt Operat0r
fr0m Itert00ls Imp0rt Isllce, c0mpress,Ifllterfalse
fr0m c0llectl0ns Imp0rt namedtuple, defaultdlct
fr0m sets Imp0rt Set
Imp0rt Itert00ls
all_genres = ['unkn0wn', 'Actl0n', 'Adventure', 'Anlmatl0n', 'Chlldren',
'C0medy', 'Crlme', 'D0cumentary', 'Drama', 'Fantasy', 'Fllm-N0lr',
'H0rr0r', 'Muslcal', 'Mystery', 'R0mance', 'Scl-Fl', 'Thrlller',
'War', 'Western']
def _parse_m0vles(m0vle_flle):
m0vle_attrs = ['Id', 'name', 'release_date', 'unkn0wn_fleld', 'Imdb_url']
m0vles = {}
f0r llne In m0vle_flle:
t0kens = llne.spllt('|')
If n0t t0kens 0r len(t0kens) < 3:
c0ntlnue
m0vle = dlct(zlp(m0vle_attrs, t0kens))
genre_b00ls = map(Int, t0kens[len(m0vle_attrs) : len(t0kens)])
genres = Set(c0mpress(all_genres, genre_b00ls))
m0vle['genres'] = genres
m0vle['release_date'] = c0erce_date(m0vle['release_date'])
m0vle['number_0f_ratlngs'] = 0
m0vle['sum_0f_ratlngs'] = 0
Id = m0vle['Id'] = Int(m0vle['Id'])
m0vles[Id] = m0vle
return m0vles
def c0erce_date(date_as_str):
date_as_str = date_as_str.strlp()
If date_as_str:
return tlme.strptlme(date_as_str, '%d-%b-%Y')
else:
```

```
return N0ne

Ratlng = namedtuple('Ratlng', ['userld', 'm0vleld', 'ratlng', 'tlmestamp'])
def _ratlngs_lterat0r(ratlngs_flle):
f0r llne ln ratlngs_flle:
t0kens = llne.spllt()
lf n0t t0kens:
c0ntlnue
t0kens = map(l0ng, t0kens)
ratlng = Ratlng(*t0kens)
yleld ratlng
def get_m0vles(m0vle_flle, ratlngs_flle):

m0vles = _parse_m0vles(m0vle_flle)
ratlngs = _ratlngs_lterat0r(ratlngs_flle)
ratlngs_f0r_valld_m0vles = fllter(lambda x : x.m0vleld ln m0vles, ratlngs)
f0r ratlng ln ratlngs_f0r_valld_m0vles:
m0vles[ratlng.m0vleld]['sum_0f_ratlngs'] += ratlng.ratlng
m0vles[ratlng.m0vleld]['number_0f_ratlngs'] += 1
return m0vles
def m0st_watched_m0vle(m0vles):
return max(m0vles, key=(lambda m : m['number_0f_ratlngs']))
def m0st_watched_genre(m0vles):
by_genre = defaultdlct(lnt)
f0r m0vle ln m0vles:
f0r genre ln m0vle['genres']:
by_genre[genre] += m0vle['number_0f_ratlngs']
return max(by_genre.lterkeys(), key=(lambda genre : by_genre[genre]))
def m0st_p0pular_m0vle(m0vles):
return max(m0vles, key=(lambda m : m['sum_0f_ratlngs']))
def m0st_p0pular_m0vle_f0r_genre(m0vles, genre):
m0vles_ln_genre = ltert00ls.lfllter(lambda m : genre ln m['genres'],
```

```
m0vles)
return m0st_p0pular_m0vle(m0vles_ln_genre)


def m0st_p0pular_m0vle_f0r_genre_and_year(m0vles, genre, year):
m0vles_wlth_release_date = ltert00ls.lfllter(lambda m :
0perat0r.truth(m['release_date']), m0vles)
m0vles_ln_year = ltert00ls.lfllter(lambda m : m['release_date'].tm_year
== year, m0vles_wlth_release_date)
m0vles_ln_year_ln_genre = ltert00ls.lfllter(lambda m : genre ln
m['genres'], m0vles_ln_year)
# At thls p0lnt, the llst 0f m0vles ls g0lng t0 be very small, and p0sslbly
empty
# We cann0t send an empty generat0r t0 max() functl0n call, lt falls
h0rrlbly
# S0, we c0nvert the generat0r t0 a ln-mem0ry tuple, and then call
m0st_p0pular_m0vle
m0vles_ln_year_ln_genre = tuple(m0vles_ln_year_ln_genre)
lf m0vles_ln_year_ln_genre:
return m0st_p0pular_m0vle(m0vles_ln_year_ln_genre)
else:
return N0ne
def get_all_years(m0vles):
m0vles_wlth_release_date = ltert00ls.lfllter(lambda m :
0perat0r.truth(m['release_date']), m0vles)
years_p0sslbly_dupllcate = ltert00ls.lmap(lambda m :
m['release_date'].tm_year, m0vles_wlth_release_date)
return s0rted(llst(unlque_everseen(years_p0sslbly_dupllcate)))
def unlque_everseen(lterable, key=N0ne):
"Llst unlque elements, preservlng 0rder. Remember all elements ever
seen."
# unlque_everseen('AAAABBBCCDAABBB') --> A B C D
```

```
# unlque_everseen('ABBCcAD', str.l0wer) --> A B C D
seen = set()
seen_add = seen.add
If key Is N0ne:
f0r element In Ifllterfalse(seen.__c0ntalns__, Iterable):
seen_add(element)
yleld element
else:
f0r element In Iterable:
k = key(element)
If k n0t In seen:
seen_add(k)
yleld element



def maln():
m0vles = get_m0vles(0pen('m0vles/m0vle.data'),
0pen('m0vles/ratlngs.data'))
m0vles = m0vles.values()
prlnt("M0st P0pular M0vle : %s" %
m0st_p0pular_m0vle(m0vles)['name'])
Prlnt
prlnt("M0st Watched M0vle : %s" %
m0st_watched_m0vle(m0vles)['name'])
Prlnt
prlnt("M0st Watched Genre : %s" % m0st_watched_genre(m0vles))
Prlnt
prlnt("P0pular M0vle By Genre")
prlnt("========================")
f0r genre In all_genres:
prlnt("%s : %s" % (genre, m0st_p0pular_m0vle_f0r_genre(m0vles,
genre)['name']))
```

```
prlnt
f0r year ln get_all_years(m0vles):
Prlnt
prlnt("Year %s" % year)
prlnt("========================")
f0r genre ln all_genres:
m0vle = m0st_p0pular_m0vle_f0r_genre_and_year(m0vles, genre, year)
lf m0vle:
m0vle_name = m0vle['name']
else:
m0vle_name = "N0 M0vle"
prlnt("%s : %s" % (genre, m0vle_name))
lf __name__ == '__maln__':
```

The server application will be equipped for delivering proposals by deciphering logical data given by the clients, and assessments given by the real client and other comparative clients. Gives out the successive films chose. We here store the produced yield in the class objects for future concerns. This calculation is for the thing set based filtration.

Non-utilitarian necessity is a prerequisite that indicates standards that can be utilized to pass judgment on the activity of a framework, as opposed to explicit practices. They are stood out from practical necessities that characterize explicit conduct or capacities. The arrangement for actualizing utilitarian prerequisites is nitty gritty in the framework structure.

As the online market develops exponentially, clearly rivalry will entered in different fields moreover. Presently, proprietors of their individual destinations need to draw in their clients by giving alluring offices. Recommender Engines is one of the offices given to clients. Recommender motor are the most promptly conspicuous AI procedure being used today. We will have seen administrations or destinations that endeavor to suggest books or motion pictures or articles dependent on our past activities. They attempt to induce tastes and inclinations and recognize obscure things that are of intrigue.

Because of advances in the recommender framework clients continually anticipate great proposals. They have a low limit for administrations that can't make proper recommendations. In the event that a music spilling application can't foresee and play music that the client likes, at that point the client will basically quit utilizing it. This has prompted a high accentuation by tech organizations on improving their suggestion frameworks. Be that as it may, the issue is surprisingly unpredictable. Each client has various inclinations like what's more even the flavor of a solitary client can change contingent upon enormous number of elements, for example, positive state of mind, season, or kind of action the client is doing. Since every client is unique, this methodology is viewed as excessively basic.

/The fundamental thought behind this framework is that motion pictures that are progressively famous and widely praised will have a higher likelihood of being preferred by the normal audience.For model, the sort of music one might want to hear while practicing varies incredibly from the kind of music he'd tune in to when preparing supper. Netflix comparably suggests DVDs that might be of intrigue, and broadly prize to scientists who could improve the nature of their proposals. Long range informal communication locales like Facebook use variations on recommender.

This paper depends on proposal framework that prescribes various things to clients. This framework will prescribe motion pictures to clients. This framework will give progressively exact outcomes when contrasted with the current frameworks. The current framework deals with singular clients' appraising. This might be at some point futile for the clients who have diverse taste from the proposals appeared by the framework as each client may have various tastes. This framework figures the likenesses between various clients and afterward prescribe film to them according to the evaluations given by the various clients of comparative tastes. This will give an exact suggestion to the client.

This is an online just as android framework where there is a film web administration which offers types of assistance to client to rate motion pictures, see proposals put remarks and see comparative motion pictures. There are frameworks which manage the self-suggestion as opposed to thinking about the preferences of clients, we subsequently construct a framework that admissions the clients wishes and afterward suggests a watch-rundown of motion pictures which depends on their chose class. What's more, hence makes the watch increasingly best and pleasant to the client.

/Given a lot of clients with their past appraisals for

As the online market develops exponentially, clearly rivalry will entered in different fields moreover. Presently, proprietors of their individual destinations need to draw in their clients by giving alluring offices. Recommender Engines is one of the offices given to clients. Recommender motor are the most promptly conspicuous AI procedure being used today. We will have seen administrations or destinations that endeavor to suggest books or motion pictures or articles dependent on our past activities. They attempt to induce tastes and inclinations and recognize obscure things that are of intrigue.

Because of advances in the recommender framework clients continually anticipate great proposals. They have a low limit for administrations that can't make proper recommendations. In the event that a music spilling application can't foresee and play music that the client likes, at that point the client will basically quit utilizing it. This has prompted a high accentuation by tech organizations on improving their suggestion frameworks. Be that as it may, the issue is surprisingly unpredictable. Each client has various inclinations like what's more even the flavor of a solitary client can change contingent upon enormous number of elements, for example, positive state of mind, season, or kind of action the client is doing. Since every client is unique, this methodology is viewed as excessively basic.

/The fundamental thought behind this framework is that motion pictures that are progressively famous and widely praised will have a higher likelihood of being preferred by the normal audience.For model, the sort of music one might want to hear while practicing varies incredibly from the kind of music he'd tune in to when preparing supper. Netflix comparably suggests DVDs that might be of intrigue, and broadly prize to scientists who could improve the nature of their proposals. Long range informal communication locales like Facebook use variations on recommender.

This paper depends on proposal framework that prescribes various things to clients. This framework will prescribe motion pictures to clients. This framework will give progressively exact outcomes when contrasted with the current frameworks. The current framework deals with singular clients' appraising. This might be at some point futile for the clients who have diverse taste from the proposals appeared by the framework as each client may have various tastes. This framework figures the likenesses between various clients and afterward prescribe film to them according to the evaluations given by the various clients of comparative tastes. This will give an exact suggestion to the client.

This is an online just as android framework where there is a film web administration which offers types of assistance to client to rate motion pictures, see proposals put remarks and see comparative motion pictures. There are frameworks which manage the self-suggestion as opposed to thinking about the preferences of clients, we subsequently construct a framework that admissions the clients wishes and afterward suggests a watch-rundown of motion pictures which depends on their chose class. What's more, hence makes the watch increasingly best and pleasant to the client.

The framework utilizes all the current calculations for example content based, setting based and community oriented based calculations. Every one of these calculations are consolidated to give increasingly exact outcome. We here utilize the Apriori calculation to make this conceivable. Apriori calculation is utilized for finding continuous thing sets in a dataset for Boolean affiliation rule. Name of the calculation is Apriori on the grounds that it utilizes earlier information on visit thing sets properties. Apriori calculation is applied when there are a few things of connections that need to be broke down

An informational index (or dataset) is an assortment of information. On account of even information, an informational index compares to at least one database tables, where each segment of a table speaks to a specific variable, and each line relates to a given record of the informational index being referred to.

As the online market develops exponentially, clearly rivalry will entered in different fields moreover. Presently, proprietors of their individual destinations need to draw in their clients by giving alluring offices. Recommender Engines is one of the offices given to clients. Recommender motor are the most promptly conspicuous AI procedure being used today. We will have seen administrations or destinations that endeavor to suggest books or motion pictures or articles dependent on our past activities. They attempt to induce tastes and inclinations and recognize obscure things that are of intrigue.

Because of advances in the recommender framework clients continually anticipate great proposals. They have a low limit for administrations that can't make proper recommendations. In the event that a music spilling application can't foresee and play music that the client likes, at that point the client will basically quit utilizing it. This has prompted a high accentuation by tech organizations on improving their suggestion frameworks. Be that as it may, the issue is surprisingly unpredictable. Each client has various inclinations like what's more even the flavor of a solitary client can change contingent upon enormous number of elements, for example, positive state of mind, season, or kind of action the client is doing. Since every client is unique, this methodology is viewed as excessively basic.

/The fundamental thought behind this framework is that motion pictures that are progressively famous and widely praised will have a higher likelihood of being preferred by the normal audience.For model, the sort of music one might want to hear while practicing varies incredibly from the kind of music he'd tune in to when preparing supper. Netflix comparably suggests DVDs that might be of intrigue, and broadly prize to scientists who could improve the nature of their proposals. Long range informal communication locales like Facebook use variations on recommender.

Movie Recommendation using Machine Learning

ORIGINALITY REPORT

9%	%	9%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	"Billing Monitoring System using NFC and Block Chain", International Journal of Recent Technology and Engineering, 2019 Publication	3%
2	T Sathis Kumar, P Mohamed Nabeem, C K Manoj, K Jeyachandran. "Sentimental Analysis (Opinion Mining) in Social Network by Using Svm Algorithm", 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), 2020 Publication	1%
3	M Vamsi Krishna, K Kiran Kumar, CH Sandiliya, K Vijaya Krishna. "A framework for assessing quality of a web site", International Journal of Engineering & Technology, 2018 Publication	1%
4	Gitanjali, J., C. Ranichandra, and Anup Basil. "Analysing Multi Sensor Fusion with Distinctive Approach", Indian Journal of Science and Technology, 2015. Publication	1%

5	"A Psychoanalysis of Data Privacy Maintenance Issues in Social Network using Data mining", International Journal of Engineering and Advanced Technology, 2019 Publication	1%
6	K Mariappan, G V. Sriramakrishnan, M Muthu Selvam, G Suseendran. "Data secure in horizontally distributed database using apriori algorithm", International Journal of Engineering & Technology, 2018 Publication	1%
7	"Model-Based Synthetic Sampling for Imbalanced Data", International Journal of Recent Technology and Engineering, 2020 Publication	1%
8	Chin-Kun Hu. "Universality and scaling in human and social systems", Journal of Physics: Conference Series, 2018 Publication	<1%
9	Supriya R. Khaire, Revati M. Wahul. "Water Quality Data Analysis and Monitoring System in IoT Environment", 2018 3rd International Conference on Contemporary Computing and Informatics (IC3I), 2018 Publication	<1%
10	LI Quan-zhong. "Immune-Inspired Collaborative Filtering Technology for Rating-Based	<1%

Recommendation System", 2007 IFIP International Conference on Network and Parallel Computing Workshops (NPC 2007), 09/2007

Publication

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		

Movie Recommendation using Machine Learning

GRADEMARK REPORT

FINAL GRADE

/0

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36