DEPARTMENT OF COMPUTER SCIENCE ENGINEERING,

SCHOOL OF ENGINEERING AND TECHNOLOGY,
SHARDA UNIVERSITY, GREATER NOIDA

MOVIE RECOMMENDATION SYSTEM

A project submitted

In partial fulfillment of the requirements for the degree of

Bachelor of Technology in Computer Science and

Engineering

By

PARANGAT NARSINGH PRADHAN (2018009122)

ADHANKI RAMACHARY (2018004863)

ABHIJEET KUMAR (2018007641)

Supervised by:

Ms. Rama M Maliya,

Asst. Professor (CSE)

May, 2022

CERTIFICATE

This is to certify that the report entitled **"Movie Recommendation System"** submitted by Parangat Narsingh Pradhan (2018009122), Abhijeet Kumar (2018007641), Adhanki Ramachary (2018004863) to Sharda University, towards the fulfillment of requirements of the degree of **"Bachelor of Technology"** is record of Bonafide final year Project work carried out by him in the **"**Department of Computer Science and Engineering, School of Engineering and Technology, Sharda University**"**.

The results/findings contained in this Project have not been submitted in part or full to

any other University/Institute for award of any other Degree/Diploma.

Signature of Supervisor

Name: Ms. Rama M Maliya

Designation: Asst. Professor (CSE)

Signature of Head of Department

Name: Prof. (Dr.) Nitin Rakesh

Place: Sharda University, India

Date: 2022/5/13

Signature of External

Examiner Date:

ACKNOWLEDGEMENT

A major project is a golden opportunity for learning and self-development. We consider our self very lucky and honored to have so many wonderful people lead us through in completion of this project.

First and foremost, we would like to thank Dr. Nitin Rakesh, HOD, CSE who gave us an opportunity to undertake this project.

My grateful thanks to **Ms. Rama M Maliya** for her guidance in our project work.

Ms. Rama M Maliya, who in spite of being extraordinarily busy with academics, took time out to hear, guide and keep us on the correct path. We do not know where we would have been without her help.

CSE department monitored our progress and arranged all facilities to make life easier. We choose this moment to acknowledge their contribution gratefully.

Name and signature of Students

PARANGAT NARSINGH PRADHAN (2018009122)

ADHANKI RAMACHARY (2018004863)

ABHIJEET KUMAR (2018007641)

Abstract

Nowadays, the recommendation system has made it quite simple to locate the items we require. There are hundreds of thousands of movies that are available to watch for the viewers but seeking a good movie out of these large numbers is very time-consuming and complex process. This is where movie recommendation system comes into play it recommends a user movie based upon their interests and preferences. Our goal with this project is to reduce human work by proposing movies depending upon a person's choice. This project proposes an engine or a system that generates suggestions on the basis of information provided by users. Movies are recommended to the user depending upon their watch history ratings and feed backs and also their psychological profile. The content-based filtering method is proposed by our system. It entails creating systems based on a user's history that is things that have already been rated. We'll be using the tmdb 5000 dataset which we obtained from Kaggle.

Index Terms Movie Recommendation System, Content- based filtering, tmdb 5000 dataset

Contents

List of Figures

List of Tables

Chapter1: INTRODUCTION

The ‘Movie Recommendation System’ is an application that can capably recommend you movies based upon your interests by taking into accounts your watch history, age, gender, etc. And also responds highly to the user’s feedbacks and ratings. The major purpose of the project is to create a ‘Movie Recommendation System’ which follows a content-based approach to understand the preferences of the user for recommendations.

The principle is to understand the user’s habits, interests their watch history, feedback, ratings to create a profile of the user and recommend movies based upon these profiles.

The ever-changing scientific world providing more and more facilities and resources to the people, recommendation system provides options to people for different things without any effort whatsoever, it saves time for people who lead a very monotonous life.

1.1 Problem Definition

People today are looking for ways to improve their lifestyles by utilizing the most up-to-date technologies available. New services that reduce human effort, save time, and simplify life are always in high demand.

People in today's modern era live very monotonous and stressful lives; they rarely have time for entertainment, and when they do, they have a plethora of options from which to choose. This is where a movie recommendation system comes into play; it generates a list of movies based on the user's preferences.

The following are the primary issues we expect to encounter in this project:

- Lack of Data: In order to research a user's preferences and make an accurate recommendation, we need a lot of data.
- Weaker Algorithms: As the number of users grows, so does the need to improve algorithm strength.

1.2 Project Purpose

Recommendation engines are tools that are used for information filtering and aim to estimate user and movie ratings, mostly through big data so as to propose their preferences. one of the major features of our system is that it helps users find movies which are most similar to the user’s taste. The major function of a recommendation engine is to identify materials for a person who may find that the materials are interesting. it also looks into various attributes to build individualized lists of content that the user may find intriguing.

Recommendation engine follows AI based algorithms that analyze all the possibilities and the output results in an individualized list of movies which may intrigue a user. The recommendations that are generated are based upon the user's profile, browsing history, search history, your likelihood of watching those films and also the activities of users who have common tastes as to that user. Heuristics and predictive modeling is used to accomplish this.

1.3 Project Overview

Movie Recommendation System or Recommendation System is a type of information filtering system that considers user data such as age, gender, as well as ratings, feedback, and watch history in order to understand the user's preferences and interests and makes recommendations based on these interests. Content-based and collaborative-based ways to designing a movie recommendation system are both viable options. Collaborative-based recommendation utilizes multiple user's interests to make recommendations to a user. We chose the content-based method because it uses a single user’s interest to generate recommendations based on that user’s preferences.Our project's main purpose is to make recommendations. When a user logs in or registers, they offer personal information such as their age, gender, preferences, and hobbies. The user's viewing history, ratings, and feedback are also taken into consideration. The user’s interests are then taken into account for generating recommendations. The interests of the user are mainly found by reviewing the ratings and feed backs that are given by the users of movies. In the end a list of recommended movies is generated that is presented to the users.

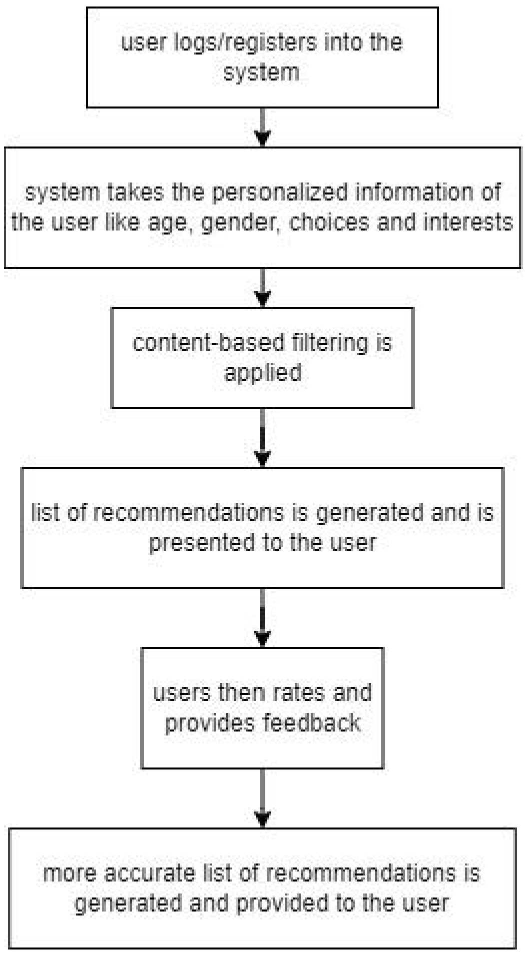

user logs/registers into the system
system takes the personalized information of the user like age, gender, choices and interests
content-based filtering is applied
list of recommendations is generated and is presented to the user
users then rates and provides feedback
more accurate list of recommendations is generated and provided to the user

Fig 1: Basic work flow of our project

our project's basic architecture/work flow is depicted in Figure 1. The following are the many steps that we took in our project:

1. Step 1: user log in or registers into the system.
2. Step 2: the system takes the personal information of the user like age, gender and preferences
3. Step 3: Content-based algorithm is applied
4. Step 4: list of recommendations is generated and is provided to the user.
5. Step 5: the user provides ratings and feed backs for better recommendations
6. Step 6: more accurate list of recommendations is generated and provided to the user.

When the list of recommendations is generated in Jupyter notebook, we use PyCharm software to create a local website where a user can type or select movies to generate recommendations, this software is mostly used by Python developers.

In PyCharm we first open a new project where we create a new python page for creating the website, then we need to import all the necessary libraries that are needed for creation of the local website and we also need to import all the necessary datasets. When all the necessary items are imported and all the coding is done, we successfully create a working local website which a user can use to generate recommendations. One of the most important libraries we have used is Streamlit. It is a highly efficient library that decreases the number of lines of code needed to complete the website.

When the local website is created, we create a login/signup page for the users and then a profile page where the user gives the necessary information like their age, gender, their interests, etc. This information of the user is very crucial for the working of the system as all the recommendations given by the system is highly dependent on this information.

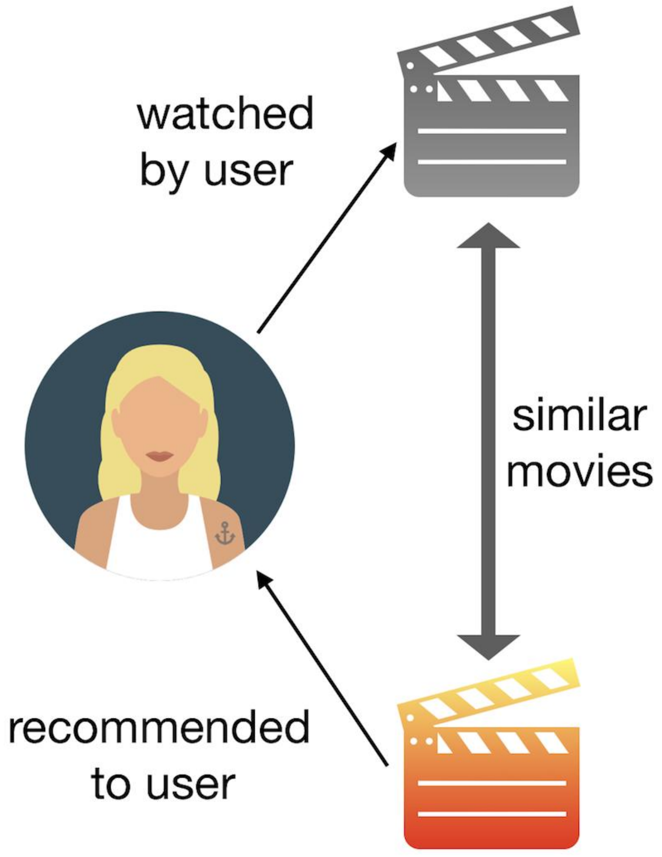


Fig 2: Features of our system

1.4 Requirement Specifications

Even the most basic of computers can run movie recommendation systems, and even an old computer may stream movies, albeit very old PCs and low-power systems such as netbooks may lag when viewing HD video. When we consider Netflix as an example of a movie recommendation system, there are a few essential characteristics that a system must meet in order to run Netflix:

1.4.1 Requirements for the network:

- A minimum speed of 1.5 Mbps is suggested.
- For standard-definition video, 3 Mbps is suggested.
- For HD video, 5 Mbps is suggested.

1.4.2 Functional Requirements:

- **Input**

The system should be able to handle the input received from the user in terms of watch history, ratings and their feedbacks, it should also be able to compare the profiles of the users with similar interests.

- **Security**

The system should be able to provide privacy and security to the data provided by the users and also be managed in a proper way.

- **Error Handling**

If the user detects any wrong name of movies, then the system should be able to give a list of movies that are similar to that name of the movie.

1.4.3 Nonfunctional Requirements:

- **Performance Requirements**

The system should be able to help users with accurate recommendations and also at very high speed.

- **Reliability**

The system should be able to gain the trust of the users by providing data privacy and security and also be reliable when providing high speed recommendations.

- **Availability**

Users should be able to access the facilities of the system from anywhere and at any time even when the system is undergoing maintenance.

- **Security**

Better provision of data security and privacy of the users and not allowing third party organizations access to the user's data. It should also provide two factor authentication to the users.

- **Ability of Learning**

It is simple to operate and reduces the learning function.

1.5 Software Specifications

Numpy
Pandas
Streamlit
Pickle
PyCharm

Table 1: Software Specifications

1.6 Language

Python

1.7 Hardware Specifications

Minimum Requirements	**Windows**
Operating System	Windows 7
Processor	single core, Intel i3
RAM	1 GB RAM
DISK Space	40 GB
Monitor	Color monitor

Table 2: Hardware Specifications

Chapter2: LITERATURE SURVEY

2.1 BASIC DEFINITONS:

Recommender System:

A recommendation system's main goal is to determines the choices of the users based upon their interests. they are being used in all most the sectors of today's era like music, news, books, movies, research articles, search queries.

Movie Recommendation System:

Movie Recommendation System is a type of information filtering system Which gives recommendations to the users based upon their gender, age, ratings, feedback and watch history. The systems study the user's input to understand their choices and preferences. Just a few examples are Netflix, Amazon Prime, Yify, and other well-known movie recommendation systems. The following are the three fundamental strategies for constructing a movie recommendation system:

- **Content-Based filtering:**

A content-based recommendation system relies on the user's input, such as ratings or search criteria. Choices are presented to the user using the user's profile which is created after taking user's input such as their ratings and search criteria. The recommendation engine becomes more precise when the user gives more input and also acts upon the recommendations provided. profile of the user's choices and description of an item are the main drivers of content-based approach.

- **Collaborative-Based filtering:**

Collaborative approach is a process of deleting items that a user may like depending upon the response of other users. its major feature is that it identifies a small group of users who have similar interests as to a single user and generates a list of recommendations by assessing their preferred movies. The model uses the user's history like their search and purchased history and also takes into account the action of other users who have very high similarity. Then it evaluates the ratings of items where the user might have an interest in.

- **Hybrid-Based filtering:**

Content-based and collaborative-based recommender systems are merged in a hybrid-based recommendation system. It combines two or more recommender system in a number of ways in order to take use of their combined features.

2.2 PROS AND CONS OF CONTENT-BASED:

Pros	**Cons**
since the recommendations are targeted towards a single customer, the system does not require information about any other customers which makes it easier for providing recommendations.	This technique necessitates a great deal of domain knowledge because the feature representation of the items is hand-engineered to some extent. As a result, the system can only be as good as the characteristics that were hand-engineered.
The system can recognize a customer's individual preferences and make recommendations for specialized things that only a few other customers are looking in.	Only current customer preferences can be used to inform the system's recommendations. In other words, the system's potential to expand on the consumers' particular interests is limited.
It does not have the 'cold start' problem that appears during Collaborative-based approach when a new user logs into the system.	Expansion is a difficult problem to solve.
It allows transparency in the recommendations provided to the user.	It's possible that certain parameters are wrong or inconsistent.

Table 3: Pros and Cons of Content-Based Approach

2.3 REVIEW:

Satya Prakash Sahu, Anand Nautiyal and Mahendra Prasad wrote a comparative study analysis in different machine learning approaches towards recommendation systems. They have used 5 different methods for experimenting on a Movie recommender system namely: content based, collaborative, hybrid, K-means clustering and Naïve Bayes. They have collected data from users based on ratings and feedback. For content-based filtering they have collected the data of a single user and given the recommendation based on the given data. For collaborative filtering they have taken the data from multiple users of similar interest and given the recommendations based upon their interest. Hybrid filtering is the result of combining content and collaborative based approach. In K-means clustering they have taken movies as clusters and have measured the proximity of the clusters by measuring the Euclidian distance between the cluster. All the clusters consist of a centroid where all the items travel towards the centroid and the centroid is updated every repetition and it stops till a saturation point is achieved. Then the users are given recommendations. Naïve Bayes takes into account the users' rating, feedback, search history and gives the recommendation based on the highest probability.

Phonexay Vilakone, Doo-Soon Park, Khamphaphone Xinchang and Fei Hao created a movie recommendation system using improved K-clique method. For creating a robust movie recommendation system, they used 6 processes: first a user is required to register into the system and enter their important details like gender, age, occupation, etc. This information was used to create experimental data and test data. The similarity between users is determined by using the experimental data. A cosine similarity measure algorithm is used to determine how similar the users are. After this a cluster of users is created based on their similarities. Then the personalized data of a unique user is compared to the personalized data of users in a cluster using cosine similarity measure algorithm. When the group has the most in similar with the new user is found then a list of movies is created where the top ones are highly rated. Then in the final process the new user is recommended with a list of movies.

Nirav Raval and Vijayshri Khedkar wrote a Review Paper on Collaborative Based approach for creating a Movie Recommendation System. This paper looks into 5 different methods for collaborative-based recommender system. They have also looked into a design methodology of movie recommender system using Neural Network. The first method is design of collaborative filtering using KNN. Here the recommender system relates 2 users based upon their ratings and recommends a movie to the users. It recommends movies to the user via their registration information. Here the ratings of users are taken into the database and the algorithms gives recommendations based upon these ratings. The second method is movie recommender system using alternating least squares (ALS). Here huge amounts of data input is taken as ratings then this data is trained and evaluated and ALS is evaluated which generates recommendations. These recommendations are stored in a SQL database and fetched using spark SQL. The third method is user-based collaborative filtering method. It is a technique of finding users with similar interests and recommending movies with the highest ratings. The first step is finding users with similar interests, this is done by using cosine similarity algorithm. The second step is predicting the rating of a movie which is yet to be rated by users of similar interests. Finally, the new user is recommended a movie based upon prediction. The fourth method is item-based collaborative filtering method. It is similar to user-based but here items with similar ratings are taken from different users. The first step is finding the similarity between different items using the cosine similarity algorithm. In the second step, a movie A which is not yet rated is compared to a similar movie B which is rated and the rating of movie A is predicted and recommended to a user. The fifth method is collaborative filtering method using K-means. First the users with similar interests are created into a cluster using cosine similarity algorithm. Each cluster consists of a centroid. Then the k nearest neighbors is found and the rating of different movies are predicted and ranked from high to low and the users are recommended movies. For design methodology of movie recommender system using Neural Network, they have taken Movie lens 1M dataset. first the dataset is split into training and testing data and the loss is filled using gradient descent. Then bias terms are added for users and movies. Then the neural network is trained and tested to give accurate recommendations of movies to the user.

F. Furtado and A. Singh created a Movie Recommendation System through the Approach of Machine Learning. They have created a recommender model by collaborative filtering method and KNN algorithm. The first step is finding users who have similar interests, this is done by using cosine similarity algorithm. Then clusters of users are created with respect to their interests. Each cluster consists of a centroid which is updated every iteration till a saturation point is achieved. The next step is choosing a neighborhood which is also created by checking clusters which have similar interests. Then the next step is predicting the rating of the movies which is not yet rated. If a new user has registered then his login data is used to find the similarity and find recommendations. They have also pointed out the drawbacks in existing models of movie recommender systems based on matrix decomposition and clustering. They have applied that combining collaborative-based filtering with Machine learning highly increases the efficiency of a movie recommender system.

Ananya Agarwal and S. Srinivasan created a movie recommender engine using hybrid approach with Apache Mahout. Hybrid filtering method is the result of combining content and collaborative based approach. here the two approaches are carried out independently and then the output of both approaches are subsequently integrated to generate recommendations. In the domains of classification, clustering, and collaborative approach, Apache Mahout provides flexible applications of complex machine learning approaches. it takes into account the user's previous ratings for making recommendations to that user, which is the major purpose of this system. They have used Yahoo Research Web scope dataset. In order to discover the resemblance between the users they have used Pearson Correlation Coefficient, higher the value of the coefficient higher the similarity between the users. The user neighborhood is found using K-nearest algorithm and similarity between movies is found using an algorithm called Log likelihood similarity. First the training data is split into training and testing data. The rating predictions on test data are then compared to the actual ratings provided in the training data. the system provides the nearest Neighbour who share the user's taste preferences and the system also recommends 5 movies to the user. It calculates the user's ratings for each movie that is recommended to that user.

2.4 EXISTING SYSTEM:

Classmates.com was the first social networking sites to rise in popularity in 1995 thanks to word-of-mouth advertising. movie industry is the area we have chosen where our idea has capabilities of making major progress over the existing processes. IMDB and AQL movies are classical movie websites which work by presenting global user ratings on the films in their library. Genre, era, directors, etc. are attributes that are used to categorize movies. these systems lack personalized recommender engine and don't exploit the different pros of social-networking networks but the users have the freedom of searching browse lists, movies and also read reviews posted by other users. Blockbuster and Yahoo movies gives recommendations based upon the user's ratings.

The popularity of companies like Netflix, whose key objective is user satisfaction, is the reason for the improvement of recommendation systems. Individuals would physically pick movies to stream from movie libraries before the recommendation system emerged. They could either read the user reviews and choose a movie based on them, or they could choose a movie at random. This procedure isn't feasible since there are so many people who have a strong preference for movies. As a result, throughout the last decade, various recommendation systems have been developed. Our current system is still under training and is facing a number of issues, as well as a low level of suggestion accuracy. The following are some of the drawbacks of our current system:

- It takes a very long time.
- Highly prone to faults
- To analyze a user's preferences and give an accurate recommendation, we need a lot of data.
- As the number of users grows, so does the increasing complexity of algorithm.

2.5 PROPOSED SYSTEM:

In this section, our proposed method for Movie Recommendation System is discussed in detail. The proposed system is summarized in Fig.3. It consists of 3 main steps importing the data set, training the dataset and testing the dataset. The output from the training model is a Movie Recommendation System. Content-based approach depends upon a single user's history. It takes into account the user's age, ratings and feedbacks to generate a list of recommendations which is presented to the user. Our system's working mechanism is broken down into the following steps:

1. Step 1: user log in or registers into the system.
2. Step 2: the system takes the personal information of the user like age, gender and preferences
3. Step 3: Content-based algorithm is applied
4. Step 4: list of recommendations is generated and is provided to the user.
5. Step 5: the user provides ratings and feed backs for better recommendations
6. Step 6: more accurate list of recommendations is generated and provided to the user.

When the list of recommendations is generated in Jupyter notebook, we use PyCharm software to create a local website where a user can type or select movies to generate recommendations, this software is mostly used by Python developers.

In PyCharm we first open a new project where we create a new python page for creating the website, then we need to import all the necessary libraries that are needed for creation of the local website and we also need to import all the necessary datasets. When all the necessary items are imported and all the coding is done, we successfully create a working local website which a user can use to generate recommendations. One of the most important libraries we have used is Streamlit. It is a highly efficient library that decreases the number of lines of code needed to complete the website.

When the local website is created, we create a login/signup page for the users and then a profile page where the user gives the necessary information like their age, gender, their interests, etc. This information of the user is very crucial for the working of the system as all the recommendations given by the system is highly dependent on this information.

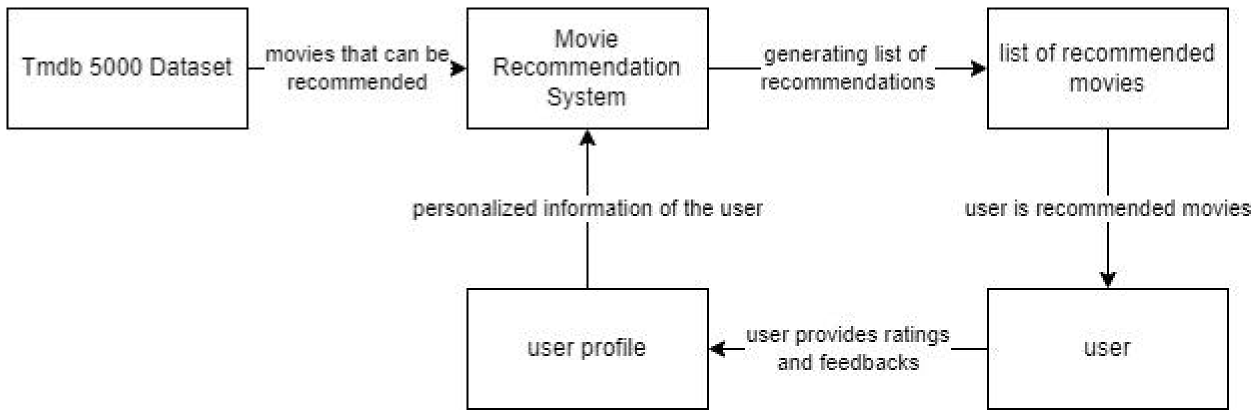

Tmdb 5000 Dataset
movies that can be recommended
Movie Recommendation System
generating list of recommendations
list of recommended movies
personalized information of the user
user is recommended movies
user profile
user provides ratings and feedbacks
user

Fig 3: Overview of our proposed system

There are four major modules in our system they are as follows:

- **Admin:**

The system admin is responsible for maintaining the system and also add movies to the data base and update it.

- **Recommendation Engine:**

It is responsible for recommending movies to the users on the basis of the users interests and preferences.

- **Web-based Movie Service:**

users will have the facility of commenting on movies and rate movies. it also gives recommendations of movies to the users.

- **User:**

The user has the facility and the freedom to rate movies and give feedback to those movies and can view movies recommended by other users.

The following are some of the benefits of our proposed system:

- Increased productivity.
- Interactive and user-friendly.
- It saves you save a lot of time.
- More data accuracy
- Improved data security for users

2.6 FEASIBILITY STUDY:

Recommender systems have made our day to day lives much easier. It influences almost all the sectors of our life from our grocery shopping to our entertainment and work. We are creating a Movie recommender engine based on Content-based approach.

Ala S.Alluhaidan has created a movie recommendation engine using collaborative-based approach but also did a comparative study on different approaches of collaborative-based method namely: user and item-based method. He concluded that user-based approach is the better approach and also paves way for the future work for the system.

1. **Operational Feasibility:**

- Our movie recommendation system will be used after the completion of system development.
- The movie recommendation system provides end users with timely results, accurate and useful recommendations of movies.
- The movie recommendation system is flexible and expandable. For future work, the system can be upgraded to include series and documentaries.
- After the complete development of the system, the user won't have to waste their time for searching movies, they will be recommended based upon their ratings, watch history and feedbacks.

2. **Technical Feasibility:**

- To operate effectively, the movie recommendation system does not require powerful computers. The system can be run on older machines, but the video and sound quality may differ. The preceding are some basic technical parameters for the system's effective functioning:
- A 1.5 Mbps network is needed for movie streaming, 3+Mbps for SD (480p) quality, 5-13+ Mbps for HD (720-1080p) quality, and 16-25+Mbps for UHD (4K,2160p) quality.
- Even the simplest computers can stream movies. below are some system requirements for streaming movies:

1. Windows 7 on a computer
2. A dual-core processor with a clock speed of 2 GHz
3. RAM: 1 GB DDR2 or faster
4. A hard disk with a rotational speed of 7200 RPM
5. Graphics card that supports DirectX 10
6. When the Movie Recommendation system is fully functional, users will be able to watch movies of varying quality depending on their system and network capabilities.

3. Social Feasibility:

checking the level of acceptance of the system is major aim of this study and also training the user on how to use the system effectively. the user should accept the system as a requirement rather than be afraid of it. The methods used to educate and familiarize the user with the system are totally responsible for the level of acceptance by the users. the constructive criticism of the users is always welcomed as he offers suggestions on how to improve the system in order to do so the user's self-esteem must be raised.

Chapter 3: SYSTEM ANALYSIS AND DESIGN

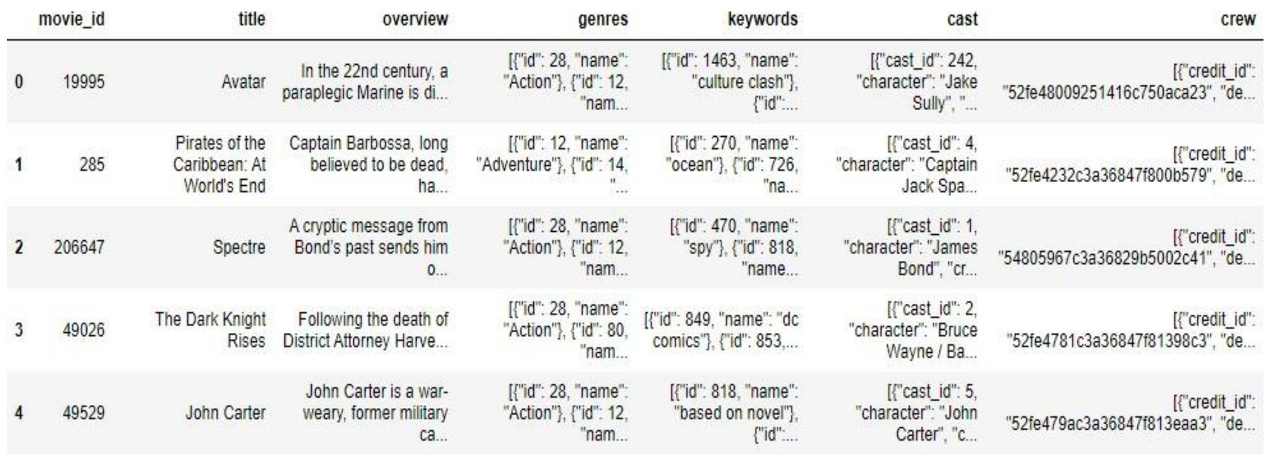
3.1 REQUIREMENT VALIDATION:

The major technological functionalities and specifications that the system should include are defined by functional requirements. The following are some of the core functions that our system must have:

- It must provide users with accurate recommendations.
- Users should be able to submit their personal data and information with confidence.
- The system should have a user-friendly interface.
- The system should run without any lags or problems.

3.2 DATASET USED:

We have used Tmdb 5000 movie dataset which we have downloaded from kaggle.com. It consists of 5000 movies, ratings, tags, budgets, credits, movie IDs, etc.



	movie_id	title	overview	genres	keywords	cast	crew
0	19995	Avatar	In the 22nd century, a paraplegic Marine is di...	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	[{"id": 1463, "name": "culture clash"}, {"id":...	[{"cast_id": 242, "character": "Jake Sully", "...	[{"credit_id": "52fe48009251416c750aca23", "de...
1	285	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	[{"id": 12, "name": "Adventure"}, {"id": 14, "...	[{"id": 270, "name": "ocean"}, {"id": 726, "na...	[{"cast_id": 4, "character": "Captain Jack Spa...	[{"credit_id": "52fe4232c3a36847f800b579", "de...
2	206647	Spectre	A cryptic message from Bond's past sends him o...	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	[{"id": 470, "name": "spy"}, {"id": 818, "name...	[{"cast_id": 1, "character": "James Bond", "cr...	[{"credit_id": "54805967c3a36829b5002c41", "de...
3	49026	The Dark Knight Rises	Following the death of District Attorney Harve...	[{"id": 28, "name": "Action"}, {"id": 80, "nam...	[{"id": 849, "name": "dc comics"}, {"id": 853,...	[{"cast_id": 2, "character": "Bruce Wayne / Ba...	[{"credit_id": "52fe4781c3a36847f81398c3", "de...
4	49529	John Carter	John Carter is a war-weary, former military ca...	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	[{"id": 818, "name": "based on novel"}, {"id":...	[{"cast_id": 5, "character": "John Carter", "c...	[{"credit_id": "52fe479ac3a36847f813eaa3", "de...

Fig4: Uncleaned version of Tmdb 5000 dataset

3.3 Block diagram, Flowchart, DFD, Use case, ER, Activity and Class Diagram:

The following is a block diagram, user-case, DFD, flowchart and ER diagram of our project, as well as the basic structure of a movie recommendation system. Below is a simple sample of how our movie recommendation system operates.

3.3.1 Block diagram:

The arrangement of operational communiques is depicted in a system architecture diagram. These are basic parts, the physical manifestations of ideas and data. The relationships between elements, characteristics, and the environment are defined by architecture. It's not easy to draw an architecture diagram.

.If a user named Ram who likes movies named Interstellar, Avatar and Gravity which are all science fiction movies then there is a very high probability that the user Ram will be recommended science fiction movies.

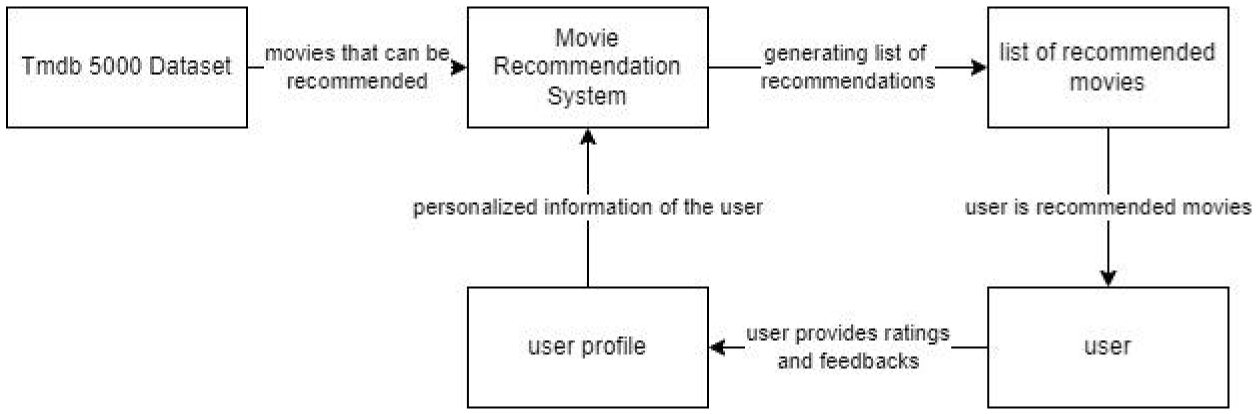

Tmdb 5000 Dataset
movies that can be recommended
Movie Recommendation System
generating list of recommendations
list of recommended movies
personalized information of the user
user is recommended movies
user profile
user provides ratings and feedbacks
user

Fig 5: Block chart of our movie recommendation system

3.3.2 Flowchart:

A flowchart is a diagram that depicts the consecutive steps of a process. It's a basic technique that may be used to define a range of processes, including project plan , manufacturing, organizational , and service operations.

- To gain a better knowledge of how a system operates
- to investigate a process in order to enhance it
- To explain how a procedure is carried out to others

In the figure below, the flowchart explains how our movie recommendation system works in detail, after importing the dataset. We read the data from the dataset and insert it into the numpy array. We then use cosine similarity algorithm to find movies that are most similar to the movies that are chosen by the user, then content based algorithm is used to create a list of movies ad is recommended to the user, the user then provides ratings and feedback of the movies and a more accurate list of recommended movies is presented to the user.

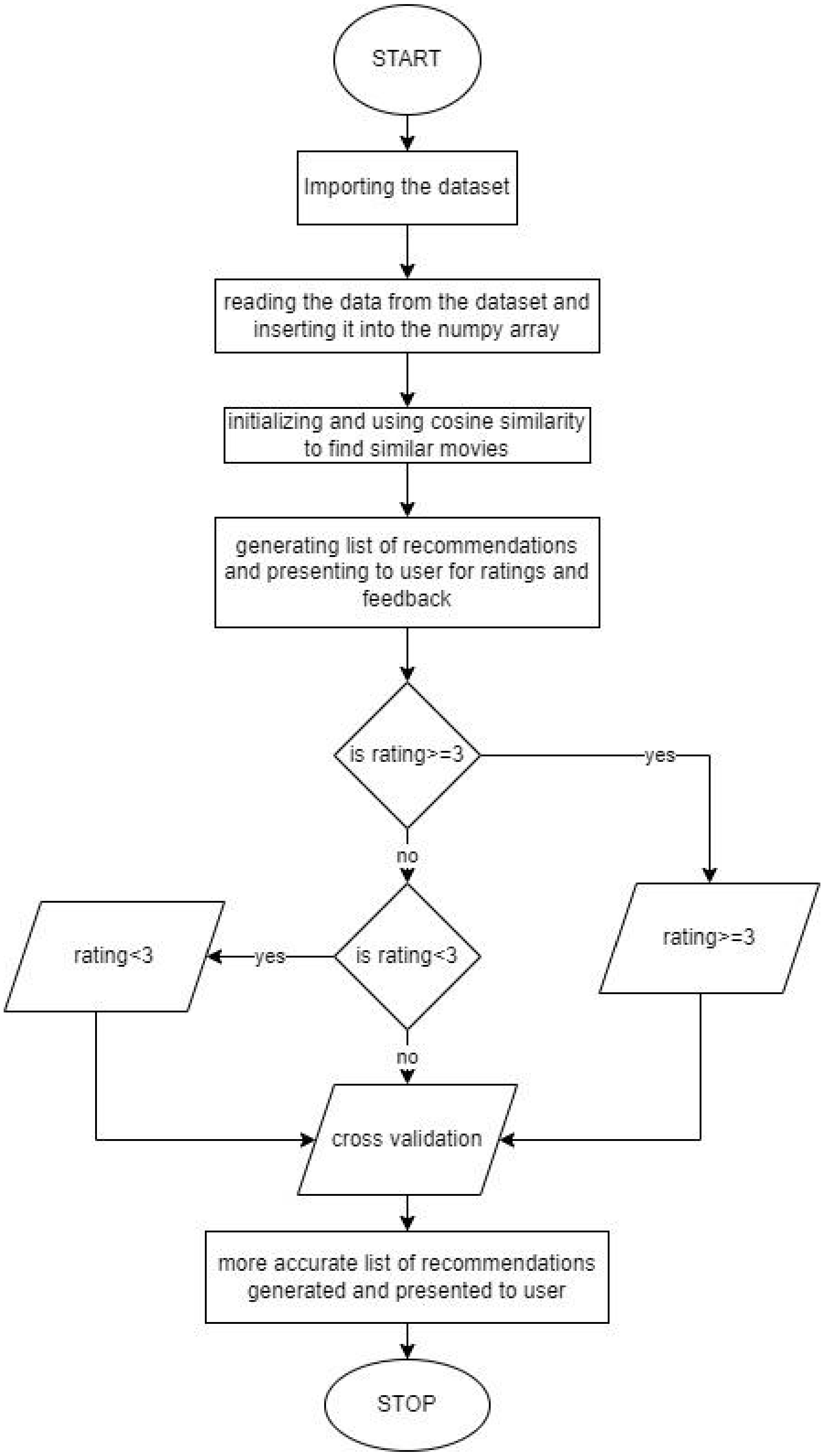

START
Importing the dataset
reading the data from the dataset and inserting it into the numpy array
initializing and using cosine similarity to find similar movies
generating list of recommendations and presenting to user for ratings and feedback
is rating>=3
yes
no
is rating<3
yes
no
rating<3
rating>=3
cross validation
more accurate list of recommendations generated and presented to user
STOP

Fig 6: flowchart of our movie recommendation system

The above flowchart describes how our system works. After the dataset has been imported and cleaned, the content-based algorithm is used to generate a list of recommended movies to the user. The user then rates and provides feedbacks of the movies and a more precise and accurate list of recommendations is generated based upon these ratings and feedbacks.

3.3.3 Data Flow Diagram:

A DFD is a classic graphic illustration of how system's information flows. DFD is used to show how information works in a system .i.e. how it enters and leaves the system. it has a purpose of determining the overall frame work of a system. It can be used to communicate between anyone else working on a system and a system analyst.

We have explained our system in three levels of DFD, they are as follows:

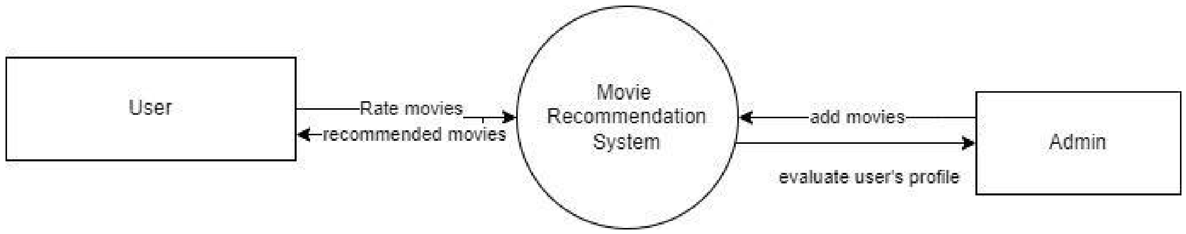


Fig 7: level 0 DFD of our system

The above fig 7 explains how our system works in a level 0 DFD. It shows in brief the relation between the user, admin and the system. This figure explains the relation between the ratings of user and the recommendations the user gets.

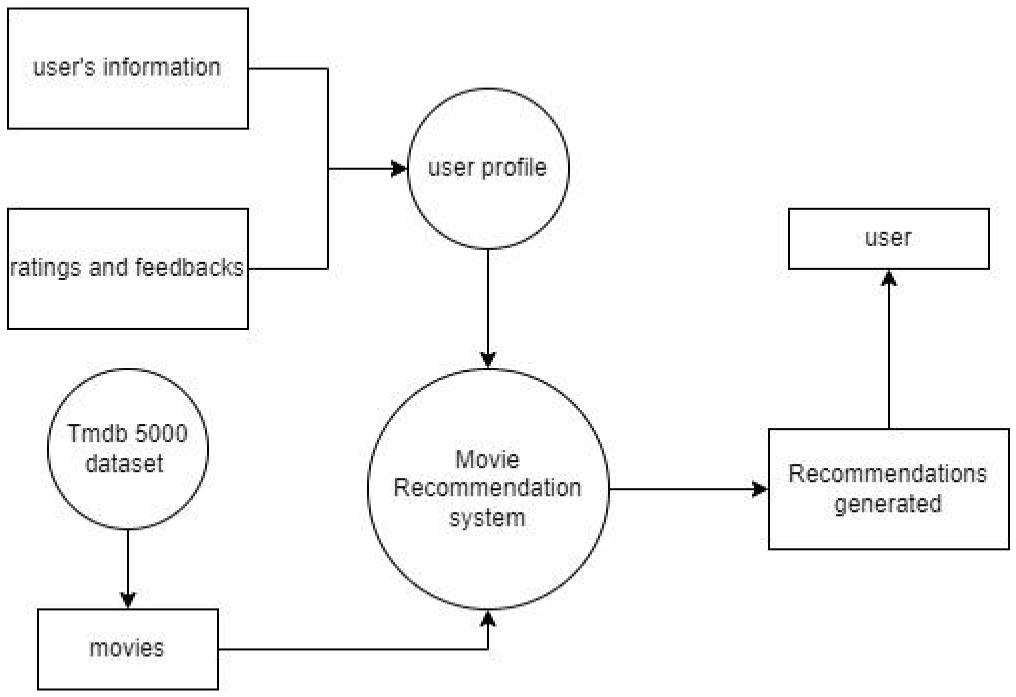


Fig 8: level 1 DFD of our system

Fig 8 shows the level 1 DFD of our movie recommendation system and it further explains the working of our system. It dives into a brief explanation of how the recommendations are generated for the user.

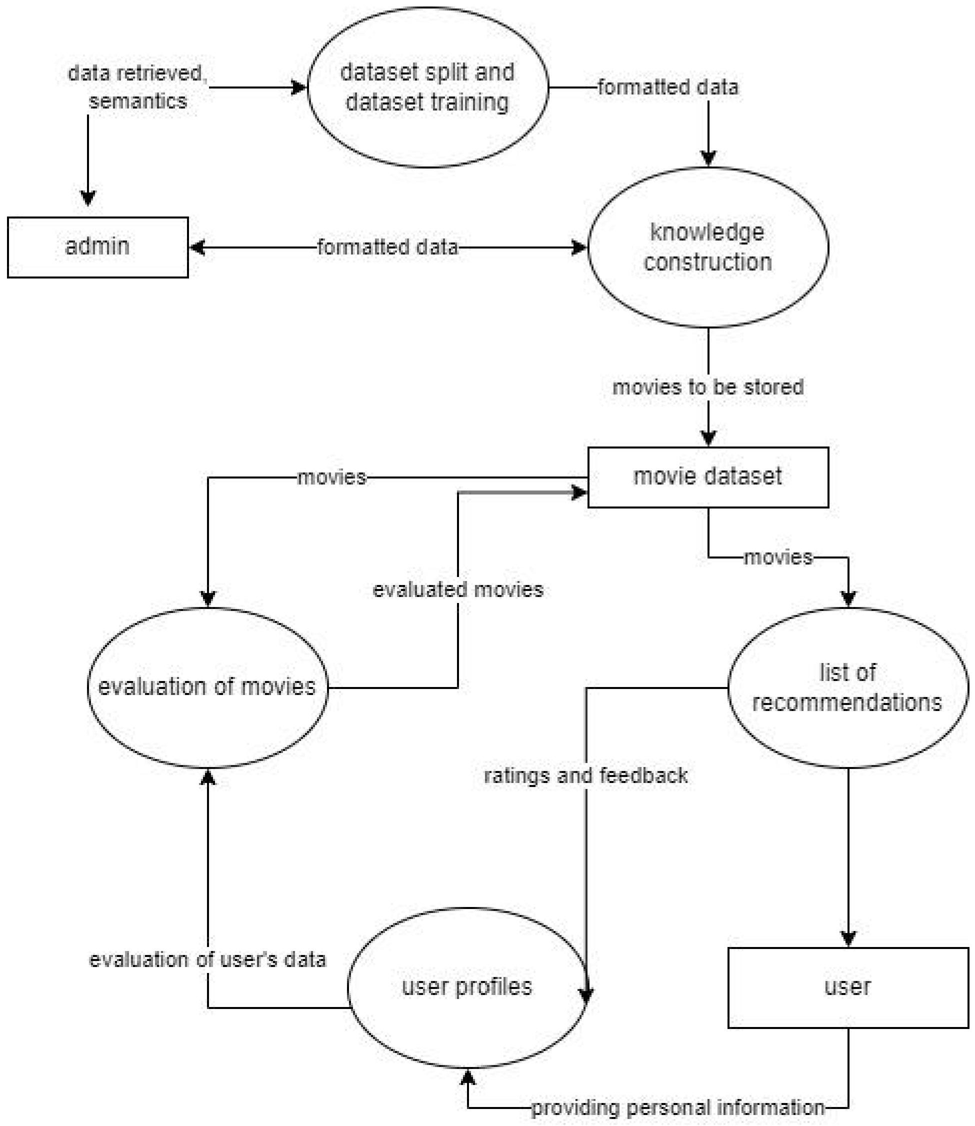


Fig 9: level 2 DFD of our system

The major pillars by which our system generates recommendations are the user's profile which is the information given by the users and ratings of movies. If a user rates horror movies highly than any other genre then the system can predict the ratings of horror movies which are unrated or unwatched by the user. The list of recommended movies is then provided to the user which the user can rate give feedback.

3.3.4 Use Case Diagram:

- Operators and use cases are used to model the operations of a program in use case diagrams. A set of services, tasks and functions that the system must do are referred to as use cases.
- list of movies together with their genre and title is provided as the input, it is the very first job in a use case.
- The degree of similarity between all pairs of movies is determined by using cosine similarity approach. it can also be called a symmetrical algorithm as the result of evaluating item A to B's and B to A is same.
- Content-based algorithm is then use to recommend movies to the user, it will give a list of recommendations to the user then the user will act on these recommendations in turn the system will be more accurate in predicting the recommendations for that particular user.
- If the user has rated a movie above 3 or 3 then the system will most likely recommend movies to that user similar to that movie.

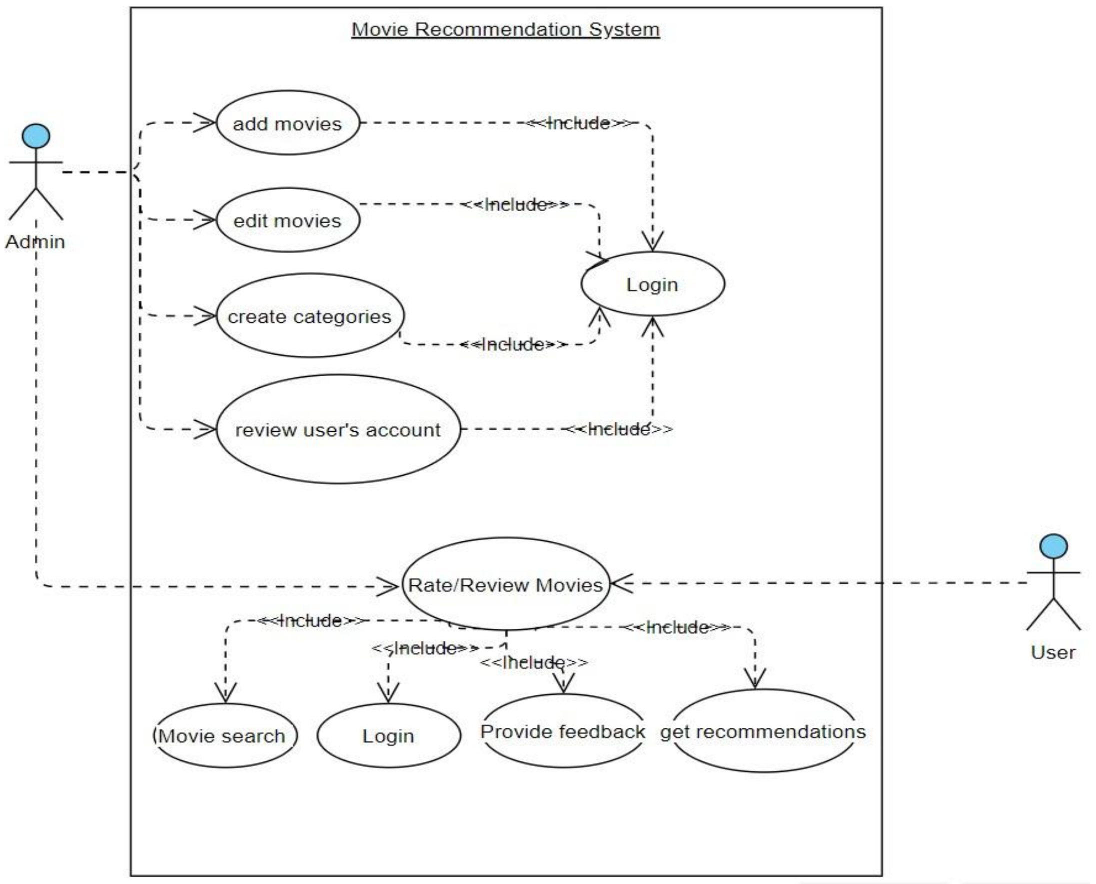


Fig 10: Use case diagram of our system

When a user who is new to the system registers into the system, a profile of the user is created which includes their age, gender, choices and when the user starts watching movies and rating them these things will be taken as input by the system and will get to know more about the user. Every time a user acts upon a recommendation the system becomes more and more accurate in giving recommendations to the user.

3.3.5 ER Diagram:

the relationship between entity sets in a dataset is shown by using Entity Relationship Diagram(ERD). It is used for understanding the logical structure of the database. the three main parts of an ER diagram are Entities, attributes and relationships.
In our tmdb5000 dataset the main entities are user, movie, genre and review. We also have used a weak entity called production company. Each of these entities have their own attributes and share different relationships among themselves. For example an entity user has attributes like gender, name, user ID, etc and shares a many to many to relationship with the other entity movie. The movie entity shares a one to one relationship with the weak attribute production company and genre entity shares one to one relationship with the entity movie. The below ER diagram explains in detail about the dataset we have used which is Tmdb 5000 dataset. There are many attributes and entities in our tmdb 5000 dataset which we don't need like production company, revenue, budget, runtime, etc. So, the first thing we do after importing our dataset is we clean our dataset and remove the unwanted items from the dataset. Then after the dataset is cleaned we can use it for training the model, a good ER diagram explains the dataset n detail which helps analyst create a robust system.

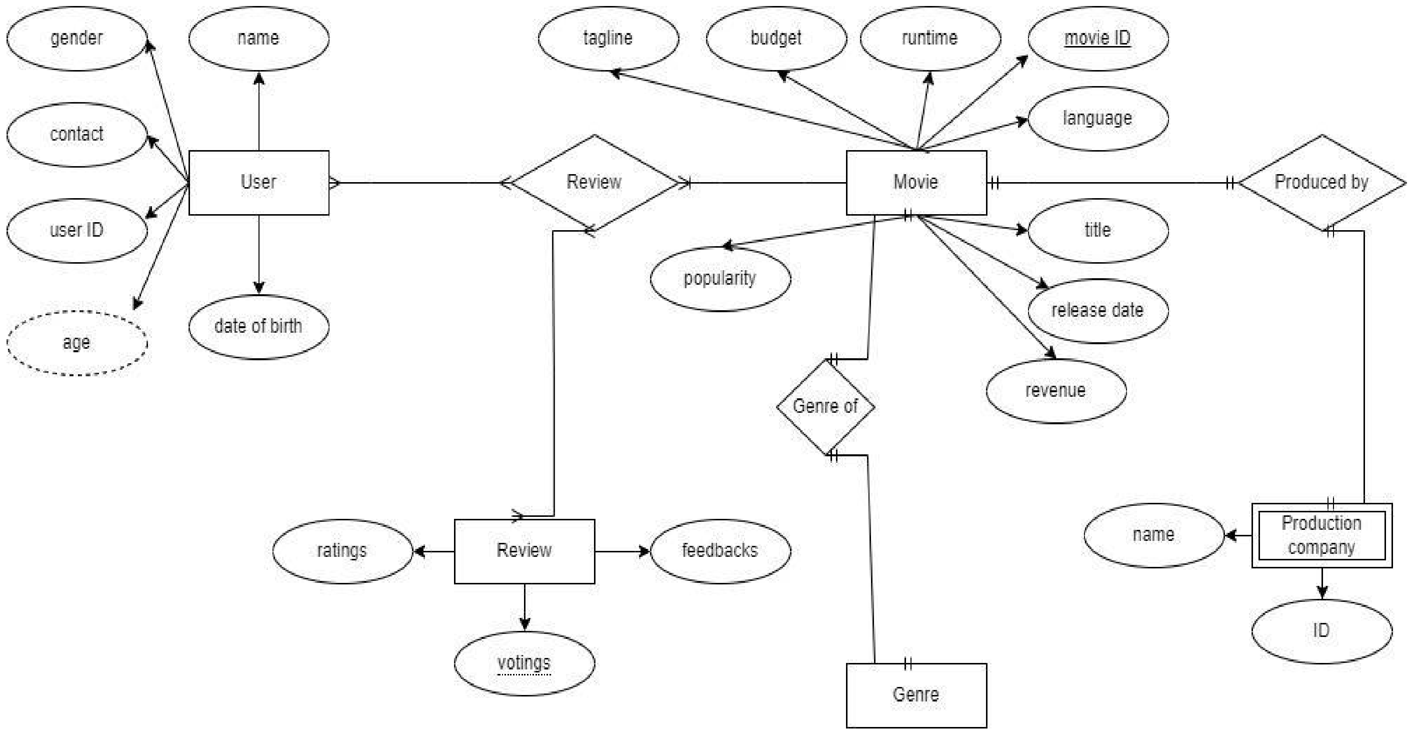


Fig 11: ER diagram of our System

A recommendation system needs some input to study the psychological profile of its user, our system creates the profile of their user based upon their feedbacks and ratings. The list of recommended movies depends heavily upon the data that is provided by the user. The suggested movies are then rated by the user. Every time the user acts on a recommendation the system becomes more accurate in generating recommendations.

3.3.6 Activity diagram:

- it is a UML diagram that does not focus on implementation but focuses on the flow of behaviour and execution of the system. object oriented flowchart is also known as activity diagram. it is also used to show software and business progression of system.
- user information is a must before you first enter.
- If the user is new user then he/she will have to register and provide their personal information like their age, gender, User ID, etc. From these information the system will create a personalized profile of the user and generate recommendations.
- The user must also choose what genre of movies they prefer.
- The system uses cosine similarity algorithm to find movies which are most similar to a particular movie.

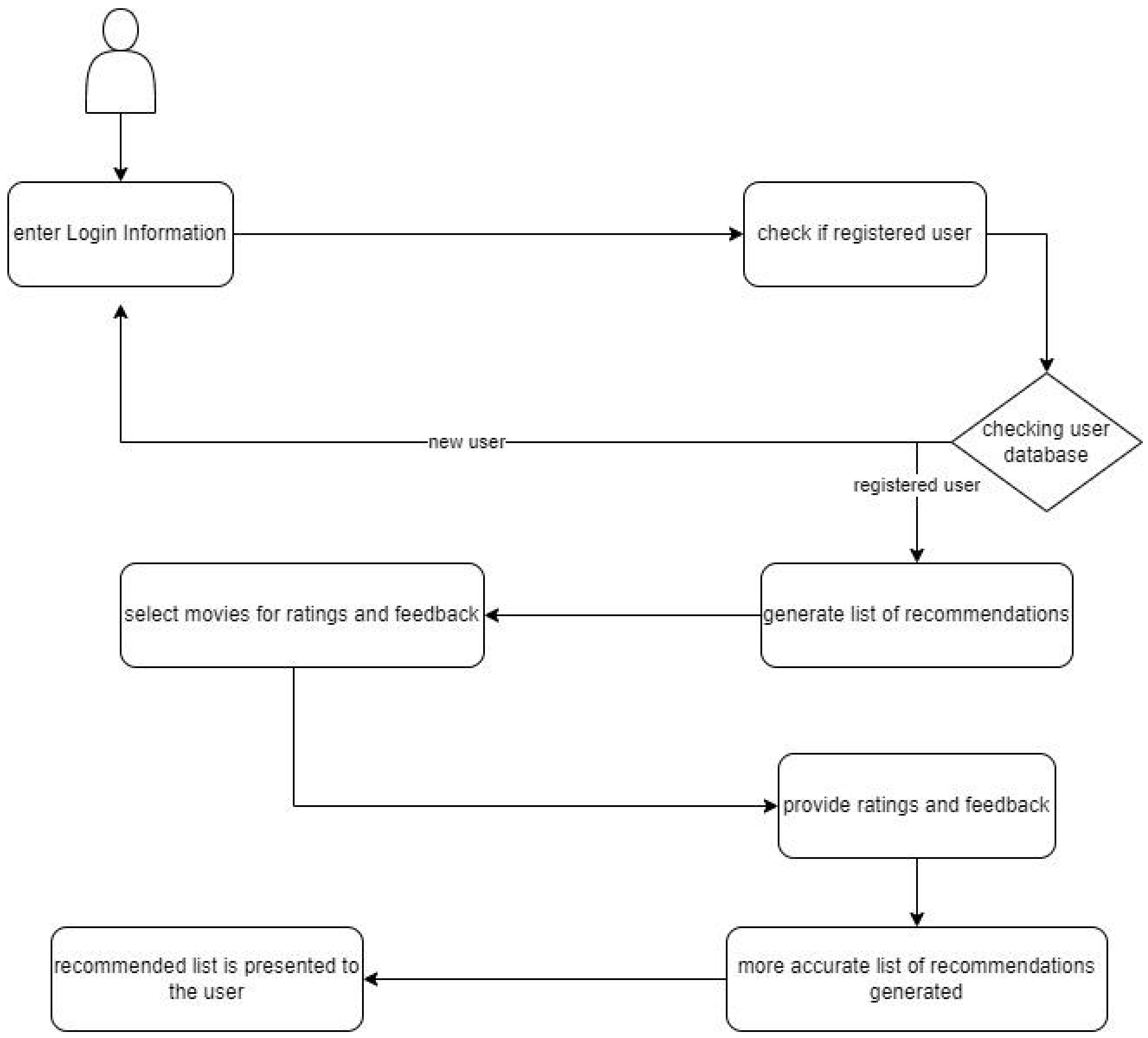

enter Login Information
check if registered user
checking user database
new user
registered user
generate list of recommendations
select movies for ratings and feedback
provide ratings and feedback
more accurate list of recommendations generated
recommended list is presented to the user

Fig 12: Activity diagram of our system

when the user logs into the system, the system will take his/her personal information such as age, name, gender, etc. To create a personalized profile of the user. When the user is presented with a list of recommendations, they will have the liberty to act in these recommendations by providing ratings and feedback and everytime a user acts in a recommendation the system will be more stronger in providing recommendations.

3.3.7 Class Diagram:

- it is a sort of static structural diagram that shows the relationships between objects, attributes and operations. It helps in understanding the system in a more detailed and elaborative point of view.
- Input movie data is a class that collects information about movies.
- Content based algorithm is for analyzing the data and generating recommendations based upon the analysis.
- Movie selection is used for selecting movies depending upon the genre of movie and ratings.
- Movie recommendation is used for recommending the user movies based upon the ratings.

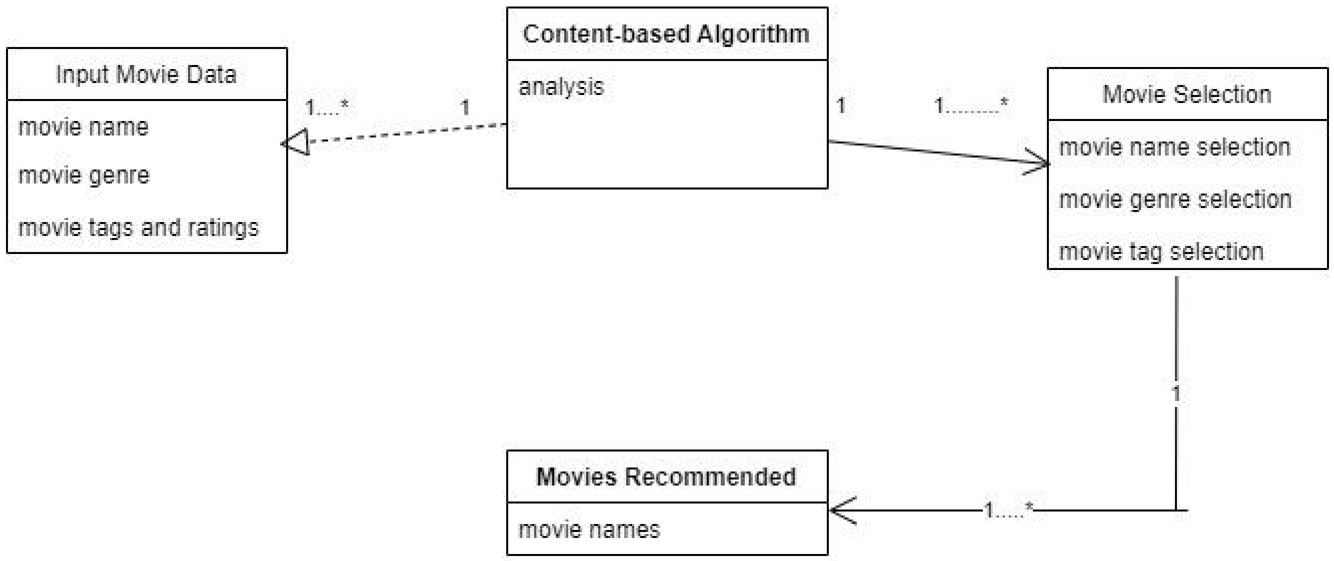


Fig 13: Class Diagram of our system

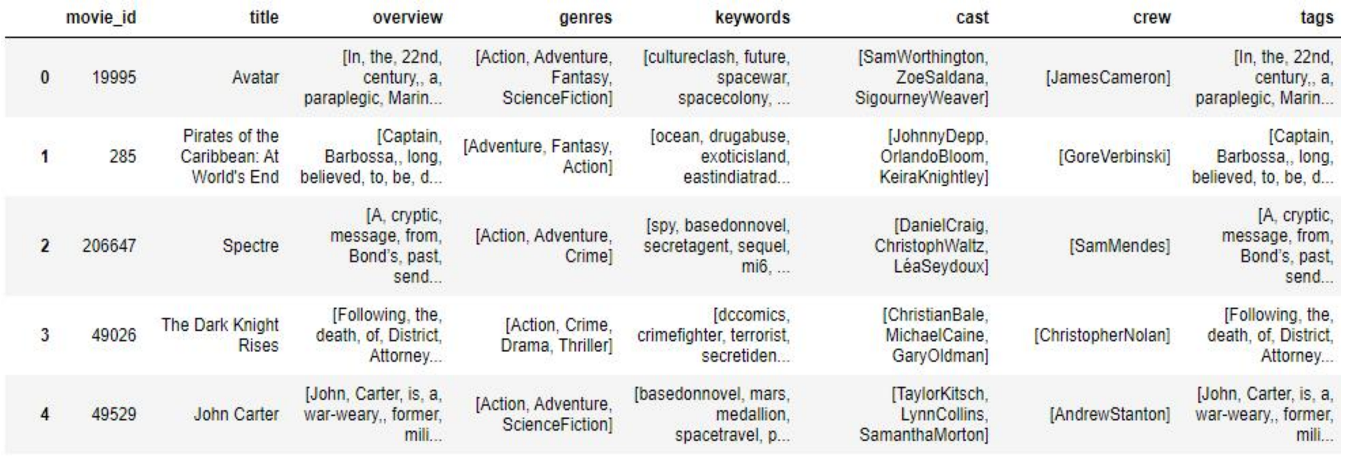
The above class diagram explains the relationship between object and entities in our system. It does not explain about the execution the system but the implementation of the system.the most important things that are needed from movies for making recommendations is movie name and genre and also rating is required. Content based filtering is used to analyse the movies to create movie recommendations that are presented to the user.

3.4 STEPS TAKEN:

Content-based approach depends upon a single user's history. It takes into account the user's age, ratings and feedbacks to generate a list of recommendations which is presented to the user. When the user rates and provides feedback or acts upon theses recommendations the system becomes more and more accurate in generating precise recommendations. Our system's working mechanism is broken down into the following steps:

1. Step 1: user log in or registers into the system.
2. Step 2: the system takes the personal information of the user like age, gender and preferences
3. Step 3: Content-based algorithm is applied
4. Step 4: list of recommendations is generated and is provided to the user.
5. Step 5: the user provides ratings and feed backs for better recommendations
6. Step 6: more accurate list of recommendations is generated and provided to the user.

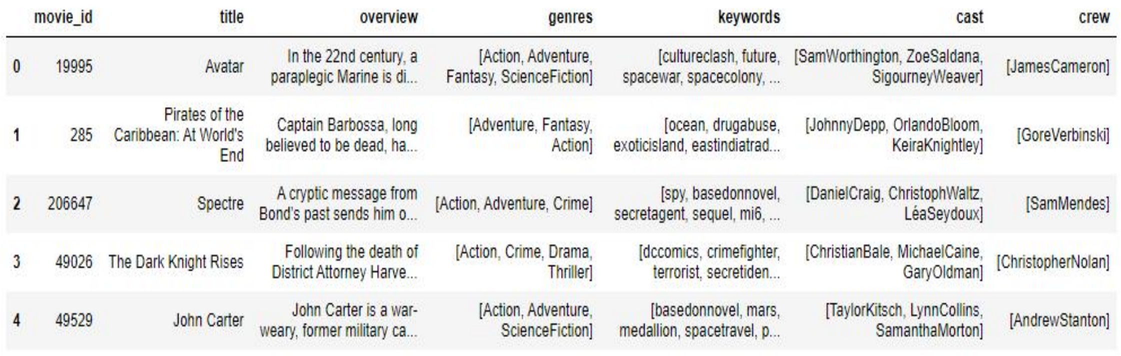
After importing the necessary libraries into the system, we import the Tmdb 5000 dataset and the dataset consists of many unwanted items like the movie's budget, revenue, status, etc. That are not necessary for generating the recommendations, so we use various functions to clean the dataset and include only those items that are crucial for generating recommendations like the movie's genre, title, language spoken, etc.



	movie_id	title	overview	genres	keywords	cast	crew	tags
0	19995	Avatar	[In, the, 22nd, century,, a, paraplegic, Marin...	[Action, Adventure, Fantasy, ScienceFiction]	[cultureclash, future, spacewar, spacecolony, ...	[SamWorthington, ZoeSaldana, SigourneyWeaver]	[JamesCameron]	[In, the, 22nd, century,, a, paraplegic, Marin...
1	285	Pirates of the Caribbean: At World's End	[Captain, Barbossa,, long, believed, to, be, d...	[Adventure, Fantasy, Action]	[ocean, drugabuse, exoticisland, eastindiatrad...	[JohnnyDepp, OrlandoBloom, KeiraKnightley]	[GoreVerbinski]	[Captain, Barbossa,, long, believed, to, be, d...
2	206647	Spectre	[A, cryptic, message, from, Bond's, past, send...	[Action, Adventure, Crime]	[spy, basedonnovel, secretagent, sequel, mi6, ...	[DanielCraig, ChristophWaltz, LéaSeydoux]	[SamMendes]	[A, cryptic, message, from, Bond's, past, send...
3	49026	The Dark Knight Rises	[Following, the, death, of, District, Attorney...	[Action, Crime, Drama, Thriller]	[dccomics, crimefighter, terrorist, secretiden...	[ChristianBale, MichaelCaine, GaryOldman]	[ChristopherNolan]	[Following, the, death, of, District, Attorney...
4	49529	John Carter	[John, Carter, is, a, war-weary,, former, mili...	[Action, Adventure, ScienceFiction]	[basedonnovel, mars, medallion, spacetravel, p...	[TaylorKitsch, LynnCollins, SamanthaMorton]	[AndrewStanton]	[John, Carter, is, a, war-weary,, former, mili...

Fig 14: dataset before preprocessing

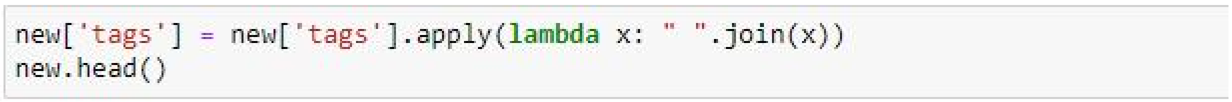
the first step we do after importing the dataset is we clean the dataset. The above dataset consists of many unwanted items like budget, revenue, status, etc. After cleaning the dataset we can proceed for training the model using this dataset.



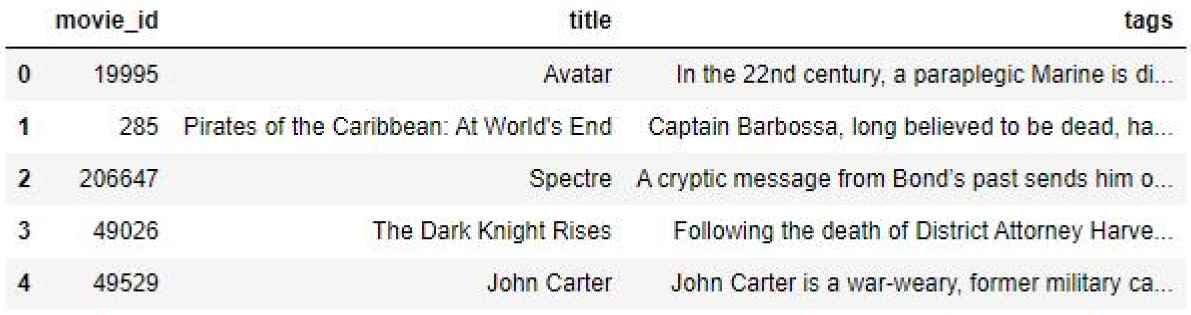
	movie_id	title	overview	genres	keywords	cast	crew
0	19995	Avatar	In the 22nd century, a paraplegic Marine is di...	[Action, Adventure, Fantasy, ScienceFiction]	[cultureclash, future, spacewar, spacecolony, ...	[SamWorthington, ZoeSaldana, SigourneyWeaver]	[JamesCameron]
1	285	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	[Adventure, Fantasy, Action]	[ocean, drugabuse, exoticisland, eastindiatrad...	[JohnnyDepp, OrlandoBloom, KeiraKnightley]	[GoreVerbinski]
2	206647	Spectre	A cryptic message from Bond's past sends him o...	[Action, Adventure, Crime]	[spy, basedonnovel, secretagent, sequel, mi6, ...	[DanielCraig, ChristophWaltz, LéaSeydoux]	[SamMendes]
3	49026	The Dark Knight Rises	Following the death of District Attorney Harve...	[Action, Crime, Drama, Thriller]	[dccomics, crimefighter, terrorist, secretiden...	[ChristianBale, MichaelCaine, GaryOldman]	[ChristopherNolan]
4	49529	John Carter	John Carter is a war-weary, former military ca...	[Action, Adventure, ScienceFiction]	[basedonnovel, mars, medallion, spacetravel, p...	[TaylorKitsch, LynnCollins, SamanthaMorton]	[AndrewStanton]

Fig 15: cleaned version of our dataset

Now we use lambda join function to merge movie_ID, title and tags



```
new['tags'] = new['tags'].apply(lambda x: " ".join(x))
new.head()
```

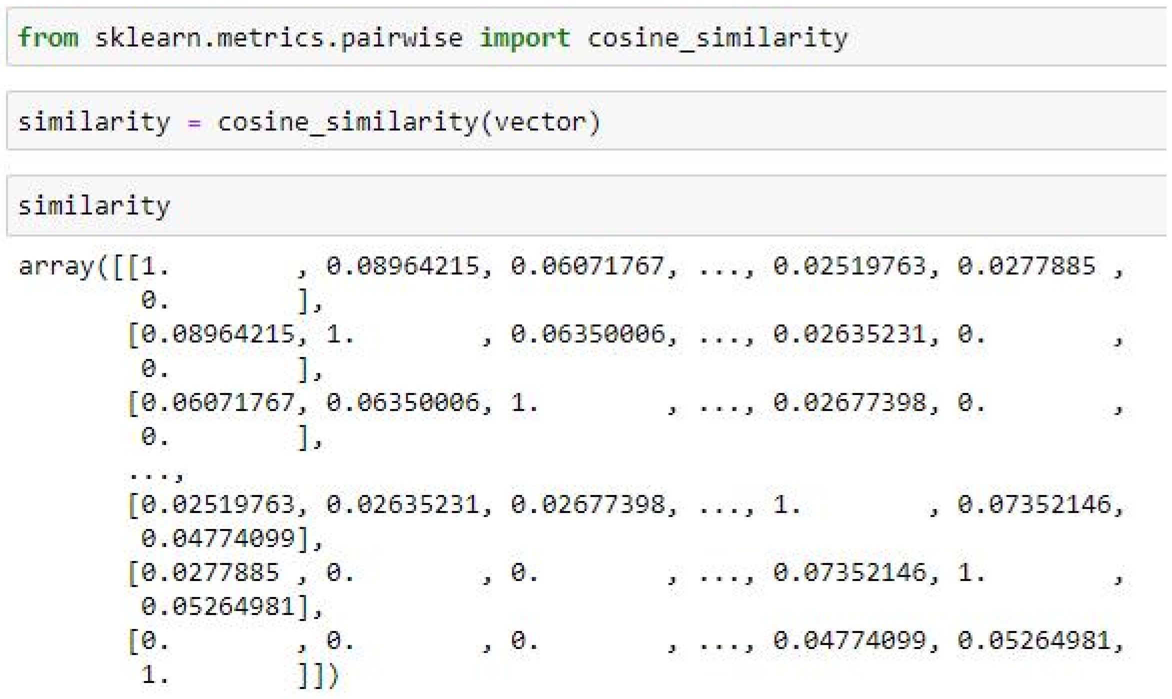


	movie_id	title	tags
0	19995	Avatar	In the 22nd century, a paraplegic Marine is di...
1	285	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...
2	206647	Spectre	A cryptic message from Bond's past sends him o...
3	49026	The Dark Knight Rises	Following the death of District Attorney Harve...
4	49529	John Carter	John Carter is a war-weary, former military ca...

Fig 16: using lambda function

Now we import CountVectorizer from Sklearn.feature_extraction to convert a given text into a vector depending on the frequency of every word that appears throughout the text

Now we use cosine similarity algorithm to find the distances between the movies, in other words to find movies that are most similar to a particular movie.The degree of similarity between all pairs of objects is determined by using cosine similarity approach. it can also be called a symmetrical algorithm as the result of evaluating item A to B's and B to A is same.



```
from sklearn.metrics.pairwise import cosine_similarity
```

```
similarity = cosine_similarity(vector)
```

```
similarity
```

```
array([[1.        , 0.08964215, 0.06071767, ..., 0.02519763, 0.0277885 ,
        0.        ],
       [0.08964215, 1.        , 0.06350006, ..., 0.02635231, 0.        ,
        0.        ],
       [0.06071767, 0.06350006, 1.        , ..., 0.02677398, 0.        ,
        0.        ],
       ...,
       [0.02519763, 0.02635231, 0.02677398, ..., 1.        , 0.07352146,
        0.04774099],
       [0.0277885 , 0.        , 0.        , ..., 0.07352146, 1.        ,
        0.05264981],
       [0.        , 0.        , 0.        , ..., 0.04774099, 0.05264981,
        1.        ]])
```

Fig 17: using cosine similarity algorithm

In Cosine similarity algorithm the similarity between two vectors decreases as the distance increases but the similarity between two vectors increases as distance decreases. So for two movies to be similar to each other the cosine distance between these two movies must be very small.
Cosine similarity and cosine distance are very common to each other. The formula to find cosine similarity is given below:

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

Fig 18: formula for cosine similarity

The formula for cosine distance is interrelated to the formula of cosine similarity. The formula for cosine distance is given by

$$1 - \text{Cosine\_Similarity} = \text{Cosine\_Distance}$$

Fig 19: formula for cosine distance

like mentioned earlier the similarity between two points increases as the distance between the point increases and the similarity decreases as the distance increases. How much cosine similarity and cosine distance is related is explained in the figure below:

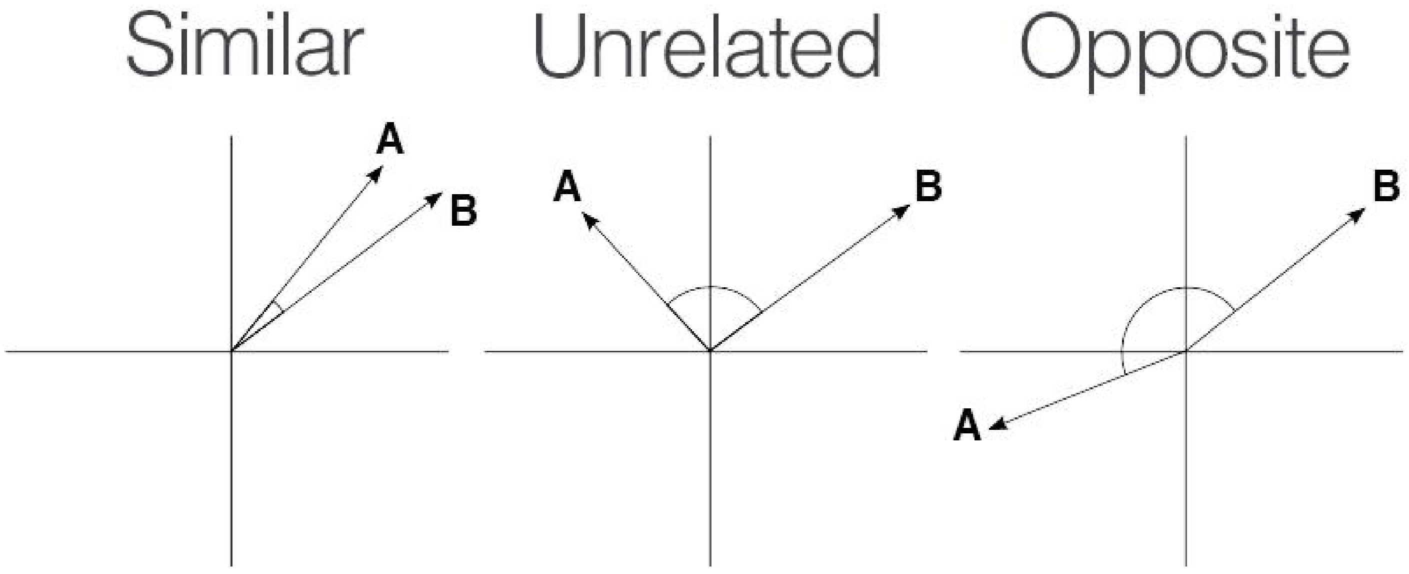
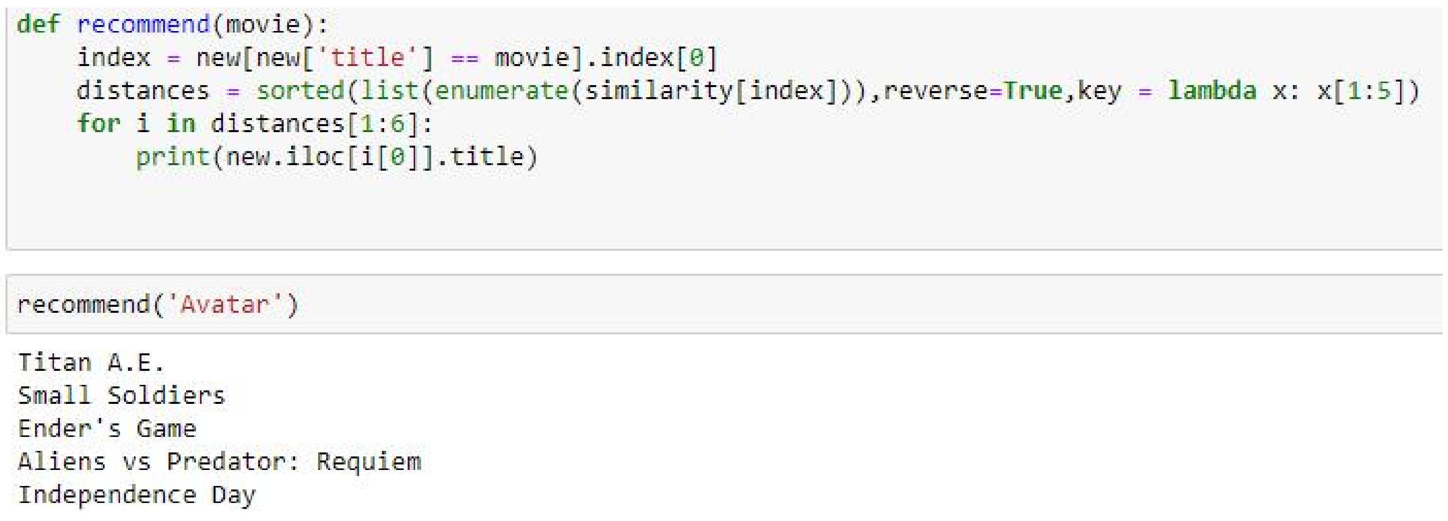


Fig 20: relation between cosine distance and cosine similarity

Then we create a function named 'recommend' and use enumerate and lambda to generate a list of recommendations. The user can type in a name of the movie in the function recommend and they will be given a list of movies.



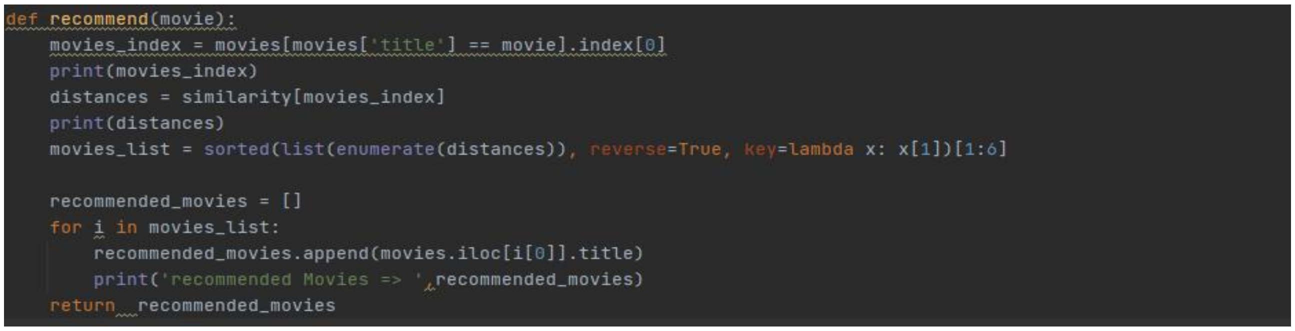
```
def recommend(movie):
    index = new[new['title'] == movie].index[0]
    distances = sorted(list(enumerate(similarity[index])),reverse=True,key = lambda x: x[1:5])
    for i in distances[1:6]:
        print(new.iloc[i[0]].title)
```

```
recommend('Avatar')
```

```
Titan A.E.
Small Soldiers
Ender's Game
Aliens vs Predator: Requiem
Independence Day
```

Fig 21: generating list of recommendations for in jupyter notebook

When the list of recommendations isgenerated in Jupyter notebook, we use PyCharm software to create a local website where a user can type or select movies to generate recommendations, this software is mostly used by Python developers.

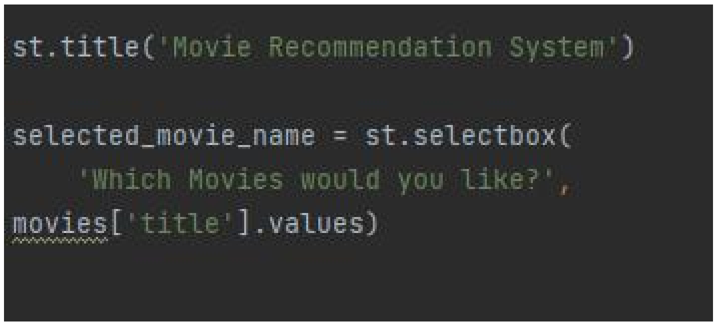


```python
def recommend(movie):
    movies_index = movies[movies['title'] == movie].index[0]
    print(movies_index)
    distances = similarity[movies_index]
    print(distances)
    movies_list = sorted(list(enumerate(distances)), reverse=True, key=lambda x: x[1])[1:6]

    recommended_movies = []
    for i in movies_list:
        recommended_movies.append(movies.iloc[i[0]].title)
        print('recommended Movies => ',recommended_movies)
    return  recommended_movies
```

Fig 22: creating functions in pycharm for generating recommendations

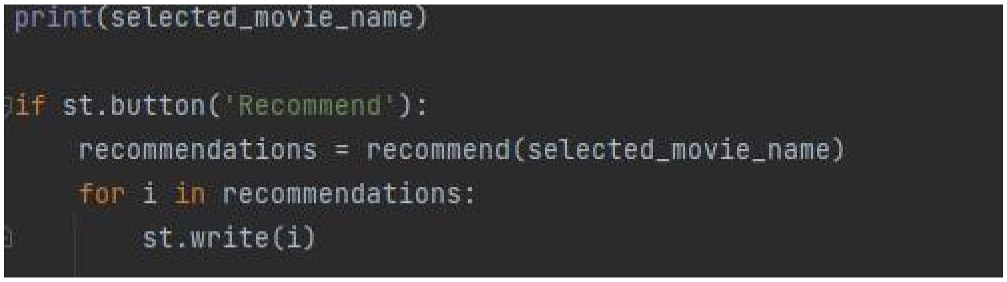
In PyCharm we first open a new project where we create a new python page for creating the website, then we need to import all the necessary libraries that are needed for creation of the local website and we also need to import all the necessary datasets. When all the necessary items are imported and all the coding is done, we successfully create a working local website which a user can use to generate recommendations. One of the most important libraries we have used is Streamlit. It is a highly efficient library that decreases the number of lines of code needed to complete the website.



```python
st.title('Movie Recommendation System')

selected_movie_name = st.selectbox(
    'Which Movies would you like?',
movies['title'].values)
```

Fig 23: creating a select box for users to select movies from

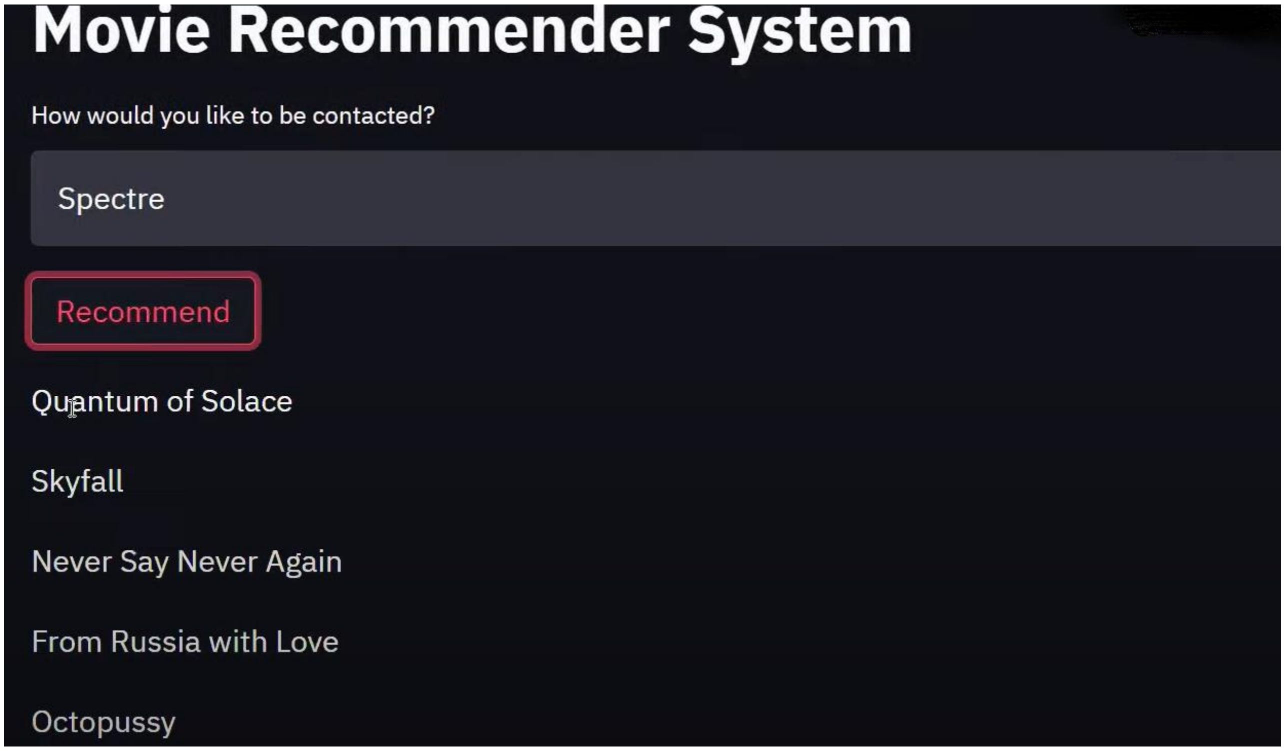


```python
print(selected_movie_name)

if st.button('Recommend'):
    recommendations = recommend(selected_movie_name)
    for i in recommendations:
        st.write(i)
```

Fig 24: writing functions to display recommended movies to the user

The recommended movies has been displayed in only textual format that is the images of the movies which are recommended has not been displayed so for this we need to import the images of the images from imdb website we can do so by using the api key from the imdb website. For generating the api key a user must log in to the website.


Movie Recommender System

How would you like to be contacted?

Spectre

Quantum of Solace

Skyfall

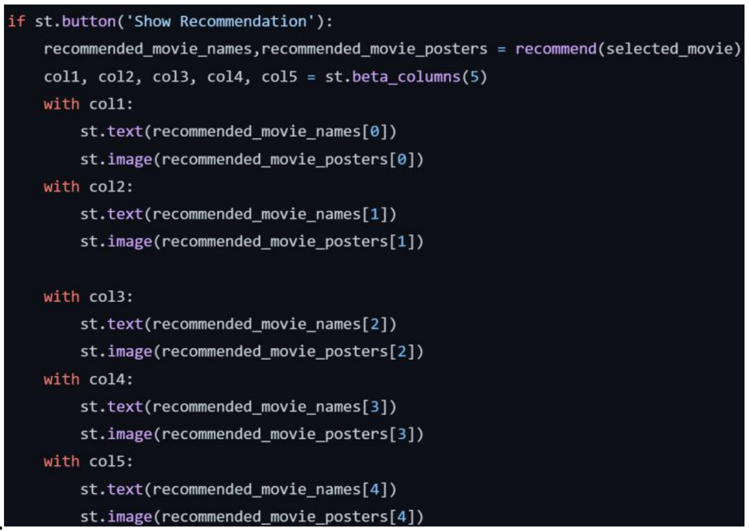
Never Say Never Again

From Russia with Love

Octopussy

Fig 25: displaying recommendations without using images

For displaying the images with the name of recommended movies we use st.image(recommended_movie_posters[]). It allows the image of the movie to be displayed under the list of recommendation


```
if st.button('Show Recommendation'):
    recommended_movie_names,recommended_movie_posters = recommend(selected_movie)
    col1, col2, col3, col4, col5 = st.beta_columns(5)
    with col1:
        st.text(recommended_movie_names[0])
        st.image(recommended_movie_posters[0])
    with col2:
        st.text(recommended_movie_names[1])
        st.image(recommended_movie_posters[1])

    with col3:
        st.text(recommended_movie_names[2])
        st.image(recommended_movie_posters[2])
    with col4:
        st.text(recommended_movie_names[3])
        st.image(recommended_movie_posters[3])
    with col5:
        st.text(recommended_movie_names[4])
        st.image(recommended_movie_posters[4])
```
.

Fig 26: code for displaying image of movie with name

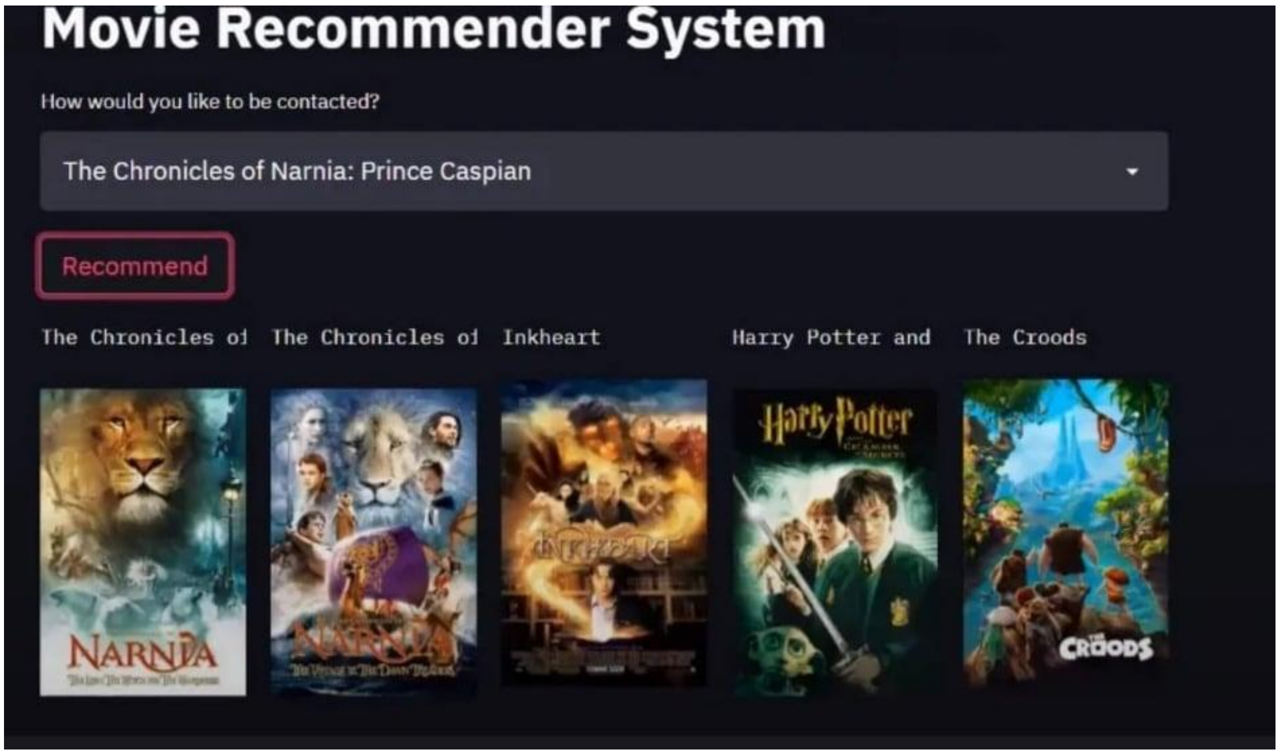


Fig 27: displaying image of movie with the list of recommended movies

When the local website is created, we create a login/signup page for the users and then a profile page where the user gives the necessary information like their age, gender, their interests, etc. This information of the user is very crucial for the working of the system as all the recommendations given by the system is highly dependent on this information.

3.5 ALGORITHMS AND TOOLS USED:

- **Content-Based Algorithm:**

A content-based recommendation system relies on the user's input, such as ratings or search criteria. Choices are presented to the user using the user’s profile which is created after taking user’s input such as their ratings and search criteria. The recommendation engine becomes more precise when the user gives more input and also acts upon the recommendations provided. profile of the user’s choices and description of an item are the main drivers of content-based approach.

- **Cosine similarity Algorithm:**

The degree of similarity between all pairs of objects is determined by using cosine similarity approach. it can also be called a symmetrical algorithm as the result of evaluating item A to B's and B to A is same.

- **Python Language:**

Python designed in 1991 by Guido Van Rossum is a very powerful high level programming language. it uses fewer lines of codes to help programmers to easily showcase their ideas and its syntax was created with code readability in view. it is being managed by Python Software Foundation.

- **Jupyterlab's notebook:**

Jupyterlab which is able to adapt easily is a development environment for using Jupyter notebook. it provides good user interface to give a large amount of data science and machine learning approaches.

- **PyCharm:**

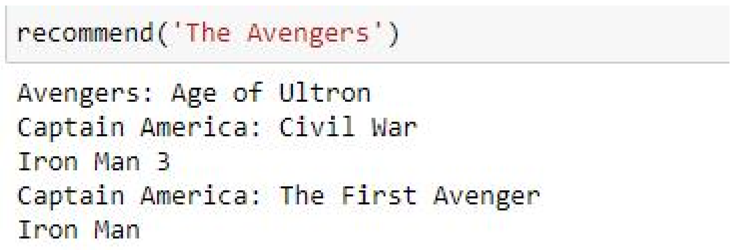
PyCharm is a Python Integrated Development Environment (IDE) that includes a variety of key tools for Python developers that are fully coupled to create a pleasant environment for effective web, python, and data science development.

- **Streamlit:**

Streamlit is a Python framework that allows you to create and share web apps for machine learning projects and data science. It is an open-source python library. With just a few lines of code, the library can help you design and launch your data science solution in moments.

Chapter 4: RESULTS / OUTPUTS

After the complete successful generation of recommendation of movies and the completion of the local website users can use can type or select movies to generate recommendations. Old and new users can sign into the system and provide their info to give their interests to the user and also provide feedback for the recommendations.



```
recommend('The Avengers')

Avengers: Age of Ultron
Captain America: Civil War
Iron Man 3
Captain America: The First Avenger
Iron Man
```

Fig 28: list generated for recommendation

Chapter 5: CONCLUSION

Movies are among the most popular forms of entertainment in today's era, and people may watch them whenever and however they want—at home, or in their cars. However, according to the normal supply and demand curve, the most popular English-language films were released 7,547 times in 2019. This movie recommendation system can be used to save time and effort when looking for a decent movie that meets our interest. Regardless of the fact that no recommendation is 100 percent accurate, the content filtering process offers a balanced set of options. Our approach takes into account the user's feedbacks and ratings. More specifications such as the movie's genre, directors and producers, actors and actress can be used in the system in the future to provide better accuracy for recommendations.

5.1 FURTHER IMPROVEMENT:

- Directors, producers and actors can be added as options for people who like particular movies of people.
- Series can also be added as recommendations for people who love to watch series, but the system needs to be trained in the same way as it was trained for recommending movies.
- Better quality of security facilities can be provided to users as the number of users increases.
- Hybrid approach can be experimented for possible and better recommendation engines., a combination of content-based and collaborative based.

Chapter 6: REFERENCES

1. “Yogesh Kumar, Ms. Naveen Kumari- Movie Recommendation System, Punjabi University Regional Centre for IT & Management, 13TH October, 2020”
2. “Sadiya Saba, Anusha S Bachihal-Movie Recommendation System Using Machine Learning, New Horizon College of Engineering,
3. Jingdong Liu, Won-Ho Choi, and Jun Liu- Personalized-Movie Recommendation Method Based on Deep Learning, Volume 2021, Article ID 6694237, 19 February 2021
4. Ala Alluhaidan, Recommender System Using Collaborative Filtering Algorithm, Grand Valley State University, April, 2013
5. Nirav Raval, Vijayshri Khedkar. -A Review Paper on Collaborative Filtering Based Movie Recommendation System, INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 8, ISSUE 12, DECEMBER 2019 ISSN 2277-8616
6. Mohit Soni and Shivam Bansa-Movie Recommendation System, Maharaja Agrasen Institute of Technology, November 13, 2019
7. Ashrita Kashyap, Sunita. B, Sneh Srivastava3, Aishwarya. PH, Anup Jung Shah-A Movie Recommender System: MOVREC using Machine Learning Techniques, SAIT, Department of Computer Science & Engineering, ISSN 2321 3361, Volume 10 Issue No.6, June 2020
8. Satya Prakash Sahu, Anand Nautiyal, Mahendra Prasad- Machine Learning Algorithms for Recommender System - a comparative analysis, University of Hyderabad, International Journal of Computer Applications Technology and Research Volume 6–Issue 2, 97-100, 2017, ISSN: -2319–8656, 14 March 2017
9. Akansh Surendran, Aditya Kumar Yadav, Aditya Kumar- Movie Recommendation System Using Machine Learning Algorithms, Raj Kumar Goel Institute of Technology, International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 07 Issue: 04 Apr 2020
10. Ananya Agarwal, S. Srinivasan- Movie Recommendation System, International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 07 Issue: 07 July 2020
11. Pradeep Kumar Singh, Pijush Kanti Dutta Pramanik, Avick Kumar Dey and Prasenjit Choudhury- Recommender systems: an overview, research trends, and future directions, National Institute of Technology, Int. J. Business and Systems Research, Vol. 15, No. 1, 2021
12. F. Furtado, A.Singh- Movie Recommendation System Using Machine Learning, Jain University, International Journal of Research in Industrial Engineering, Int. J. Res. Ind. Eng. Vol. 9, No. 1 (2020) 84–98
13. Phonexay Vilakone, Doo-Soon Park, Khamphaphone Xinchang1 and Fei Hao- An Efficient movie recommendation algorithm based on improved k-clique, Soonchunhyang University, Vilakone et al. Hum. Cent. Computer. Inf. Sci. (2018) 8:38

14. Bela Gipp, Jöran Beel, Christian Hentschel- A Research Paper Recommender System, Fraunhofer Institute for Telecommunications, PaperID: 213, 21 May 2014.
15. Hulong Wang, ZeshengShen, Shuzhen Jiang, Guang Sun, Ren-jie Zhang- User-based Collaborative Filtering Algorithm Design and Implementation, Hunan University of Finance and Economics, ICCBDAI 2020 doi:10.1088/1742-6596/1757/1/01216