MOVIE RECOMMENDATION SYSTEM

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering

By

Tarun Soni (191304)

Under the supervision of

Dr. Amol Vasudeva (Assistant Professor, CSE)

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology

Waknaghat, Solan-173234, Himachal Pradesh

Candidate's Declaration

I hereby declare that the work presented in this report entitled **"Movie recommendation system"** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from January 2023 to May 2023 under the supervision of **(Dr. Amol Vasudeva)** (Assistant Professor, CSE).

I also authenticate that I have carried out the above-mentioned project work under the proficiency stream Cloud Computing.

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

(Student Signature)
Tarun Soni, 191304

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

(Supervisor Signature)
Dr. Amol Vasudeva
Assistant Professor
CSE
Dated:

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: ....

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

Name: ________________________ __**Department:** _______________ **Enrolment No** ________

Contact No. ___________________________**E-mail.** __________________________________

Name of the Supervisor: ____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): _____________________

____

____

UNDERTAKING

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

Complete Thesis/Report Pages Detail:

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

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at ....(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

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

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

Checked by
Name & Signature **Librarian**

....

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

ACKNOWLEDGEMENT

All compliments and praise are due to God who empowered me with strength and sense of devotion to successfully accomplish this project work successfully.

I am really grateful and wish my profound indebtedness to Supervisor Dr Amol Vasudeva, Assistant Professor (Grade II), Department of CSE Jaypee University of Information Technology, Waknaghat. Deep Knowledge & keen interest of my supervisor in the field of "cloud computing" to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

I would also generously welcome each one of those individuals who have helped me straightforwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

Tarun Soni
191304

Table of Contents

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Recommendation systems have become an essential component in various industries such as e-commerce and OTT-platforms. These systems use various algorithms to recommend the most relevant data to the user. Movie recommendation systems, in particular, suggest movies based on the user's interests, thus saving time and effort for the user in searching through a large list of movies to watch. The aim of this project was to develop a movie recommendation system using cosine similarity algorithm. The system is designed to provide personalized movie recommendations based on the user's movie preferences. The project began with data collection from various sources, including movie reviews, ratings, and user preferences. The collected data was preprocessed and transformed into a structured format suitable for analysis. The development of a cosine similarity algorithm comes next. This algorithm is used to compare two sets of vectors. The technique was used to assess how well the films in the dataset fit the user's preferences. The method for proposing films was built using a web-based interface. The interface allows users to enter their film tastes and receive suggestions based on what they say. The suggestions are presented in descending order of similarity, with the most comparable films at the top. Even films that the user has never heard of could be suggested by the system. Giving users a wide range of recommendations that are tailored to their particular preferences is made possible thanks to this capability. In conclusion, the project's goal of developing a cosine similarity-based movie recommendation system was accomplished. Users could easily access and interact with the system because of its web-based interface, and it was quite accurate at providing individualized recommendations.

CHAPTER 1: INTRODUCTION

1.1 Introduction

The systems that make recommendations to users based on a variety of parameters are known as recommender systems. The recommender system handles the abundance of information by filtering the most crucial information based on the information provided by a user and other criteria that take into account the user's choice and interest. It determines whether a user and an item are compatible and then assumes that they are similar in order to make recommendations. Thus, The user's tastes and conduct are taken into account when making the recommendation, and the best options are then shown to them.

Movies have a crucial role in life. There are many different kinds of movies, such as those made for amusement, those meant to be instructive, documentaries, horror films, animated cartoons, and action movies. These can be easily distinguished based on their genres, such as action, animation, comedy, thriller, emotional, horror, and sci-fi. There are many methods to differentiate between different movies, including by their release year, language, director, etc. The problem of wasting a lot of time looking for our favourite films to watch is reduced thanks to movie recommendation systems' assistance in finding our preferred films among all the available sources. Therefore, it is crucial that the system used to recommend films be very accurate and trustworthy so that it can give us recommendations for films that are either exactly the same as our preferences or most closely match them. Since recommendation systems have advanced over the past 20 to 25 years, users have come to expect high-quality recommendations. A user will stop using a video streaming app if it cannot recognise and play videos they will enjoy. This prompted tech firms to enhance their recommendation systems. Due to the fact that each person has various tastes and likes, the issue is not as straightforward as it first appears to be. Additionally, a single user's choice can change depending on a wide range of variables, including the time of year, their current mood, and occasionally the activity they are engaged in. For instance, the music one might choose to listen to while working out in the gym differs significantly from the music one might prefer to listen to when cooking. The exploration vs. exploitation conundrum should also be resolved by the recommendation system. While still maximising what is previously known about the user, the recommendation engine needs to investigate more domains to learn more about the choices and preferences of users.

1.1.1 Recommendation system work mechanism

Machine learning techniques and specialized algorithms are used in recommendation systems. The recommendation system is powered by automated configuration, coordination, and management of machine learning predictive analytics algorithms, allowing it to make informed decisions about which filters to use depending on the circumstances around a given user. It enables marketers to raise conversion and order value averages.

Recommender systems are useful tools because they can predict user ratings before the users themselves do. A recommendation system primarily processes data through the following four stages:

1. Data collection: Information may be implicit (such as page visits and order histories) or explicit (such as ratings and comments on specific products).
2. Storing: Whether you should use a NoSQL database, object storage, or a regular SQL database for storage depends on the type of data utilised to generate the recommendations.
3. Analysing: Following analysis, the recommender system identifies things with comparable user interaction data.
4. Filtering: This is the final phase in which data is filtered to obtain the pertinent data needed to offer user recommendations. You must select an algorithm that works with the recommendation system in order to activate this.

1.1.2 Types of recommendation systems

Although there are many issues that machine learning can solve, product recommendations are one of its most well-known uses. Three primary categories of recommendation systems exist:

i. Collaborative Filtering

Collaborative filtering is based on collecting and examining user behaviour data. This includes anticipating what the user would appreciate based on similarities to other users and the user's online activity.
They share interests, for instance, if user A like apples, bananas, and mangoes while user B enjoys apples, bananas, and jackfruit. It is therefore very likely that A will like jackfruit and B will like mango. Collaborative filtering operates in this manner. Two kinds of collaborative filtering techniques used are:

- User-User collaborative filtering
- Item-Item collaborative filtering

This recommendation system's ability to make precise recommendations without having any prior knowledge of the recommended item is one of its primary features. There is no reliance on content that is machine-processable.

ii. Content-Based Filtering

The description of a product and a profile of the user's chosen options are the foundations of content-based filtering techniques. A user profile is created in this recommendation system to convey the types of things the user is interested in. things are characterised using keywords.
For instance, if a user enjoys watching films like Iron Man, the recommender system will suggest other superhero films or films that focus on Tony Stark.

The fundamental premise of content-based filtering is that if you enjoy one item, you'll likely enjoy a related item as well.

iii. Hybrid Recommendation Systems

To provide clients a wider selection of products, hybrid recommendation systems promote products using both content-based and collaborative filtering simultaneously. This emerging recommendation system is rumoured to deliver recommendations that are more accurate than those made by other recommender systems.

A great example of a hybrid recommendation system is Netflix. By comparing users' viewing and searching patterns and identifying users who use the platform similarly, it generates suggestions. Netflix use collaborative filtering in this manner.

Netflix employs content-based filtering by suggesting similar series and films to those that the user has rated highly. Additionally, they have the power to reject problems with cold starts and data scarcity that are frequent in recommendation systems.

1.1.3 Genre based filtering

Genre-based filtering is a type of content-based algorithm that proposes films based on the user's chosen genre. The algorithm analyses the genres of the movies the user has already viewed and enjoyed before making recommendations for other movies in that genre.

A movie recommendation system can add genre-based filtering by following the procedures outlined below:

1. Gather information: Compile data about the films that are available on the system. This part should include the genre of each movie as well as any other essential information, such the actors, director, story, and keywords.

2. User profile management: A profile should be established for each user in the system. In this profile, the user's choices, particularly their favourite genres, should be made clear.

3. Analyse the genre of the films the user has already seen and enjoyed. Natural language processing methods or the metadata tags that specify the movie's genre can be utilised to extract the keywords and themes from the movie descriptions.

4. Generating recommendations for the user using the knowledge about genres. This can be accomplished by supplying suggestions for films in genres similar to those the user has already seen and liked.

5. Updating the data: As the user provides input on the suggestions, make adjustments as needed. This can be accomplished through collaborative filtering approaches, where the system examines the preferences and behaviour of other users to generate recommendations that are more accurate.

6. Presentation: Make the recommendations approachable for the user by presenting them as a list of suggested films that can be sorted and filtered by genre.

All the above steps are important in order to make a good working movie recommendation system in the said manner. We have also tried to follow the same procedure in the following project.

1.1.4 Types of Genre based algorithms

Here are some examples of algorithms that can be used as genre-based filtering for movie recommendation systems:

1. K-Nearest Neighbor (KNN) algorithm: The k-nearest neighbours algorithm is a non-parametric, supervised learning classifier that relies on closeness to classify or anticipate how a particular data point will be grouped. Although it can be applied to classification or regression issues, it is commonly employed as a classification algorithm because it relies on the idea that comparable points can be discovered close to one another. Utilising this method, it is possible to create films with similar genres, casts, directors, and plots. The KNN algorithm locates the K closest neighbours of the user's favourite films and suggests further films that are comparable to these neighbours.

2. Decision Tree algorithm: It is a supervised machine learning tool that may be used to classify or forecast data based on how queries from the past have been answered. The model is supervised learning in nature, which means that it is trained and tested using data sets that contain the required categorisation. The decision tree might not always offer a simple solution or choice. Instead, it might give the data scientist choices so they can choose wisely on their own. Decision trees mimic human thought processes, making it generally simple for data scientists to comprehend and evaluate the findings. This algorithm creates a decision tree that connects the recommended recommendations to the user's favourite genre. It creates a decision tree that suggests films based on the user's preferences after analysing the user's viewing history and favourite genres. The decision tree algorithm might suggest further horror films with comparable attributes if, for instance, the user has viewed and appreciated a number of horror films.

3. Support Vector Machine (SVM) algorithm: A supervised machine learning approach called Support Vector Machine (SVM) is used for both classification and

regression. Although we also refer to regression issues, categorization is the most appropriate term. Finding a hyperplane in an N-dimensional space that clearly classifies the data points is the goal of the SVM method. The number of features determines the hyperplane's size. The hyperplane is essentially a line if there are just two input features. The hyperplane turns into a 2-D plane if there are three input features. It becomes difficult to imagine when the number of features exceeds three. This system, which divides movies into many genres, uses machine learning. It evaluates the film's elements, including the cast, director, and plot, and categorises each film into one or more genres. The SVM algorithm suggests films that are categorised in the same genre as the user's favourite genre.

4.. Matrix Factorization algorithm: This algorithm uses matrix factorization techniques to recommend movies based on the user's preferred genre. It analyzes the user's viewing history and generates a matrix that represents the user's preferences for each genre. Based on this matrix, the algorithm recommends movies that are similar to the viewer's preferred genre.

These are just a few examples of genre-based filtering algorithms for movie recommendation systems. Each algorithm can be used alone or in conjunction with other algorithms to give viewers with tailored movie suggestions based on their tastes and viewing history. Each algorithm has benefits and disadvantages.

1.2 Problem Statement

The primary goal of the "Movie Recommendation System" project is to suggest a movie in line with the user's preferences, i.e., by giving the user the relevant content from irrelevant collections of unrelated objects.

In the statement of the concern, providing the relevant content to the user’s out of the related and the insignificant collections of the items, the main purpose of this project “Movie Recommendation System” is to suggest or recommend the movie to the user’s according to the user-specific movie ratings. Looking at the set of the users having their past movie ratings, can we say the rating they will give to the movies is not rated already? Let us take an example “Which film would you like”, if you look at “spiderman 1”, “spiderman 2”, the users who saw these films also liked “spiderman 3”? The system will take into account a user's past interactions with movies, including their ratings and reviews, to generate personalized recommendations.

1.3 Objectives

The main objective is to offer customers personalised movie recommendations based on their tastes, interests, and prior viewing habits. Genre-based movie recommendations to the user are one approach to accomplish this. The system can make recommendations for films that are likely to be of interest to the user by examining the user's movie tastes. For instance, if a user frequently watches action films, the system may be able to suggest further action films the user might like. Similar to how it can offer more romantic comedies if the user is interested in that genre. A movie recommendation system may also aim to improve user experience by developing a website that is simple to use and browse.

The overall goal of the movie suggestion system is to give users a unique and satisfying movie-watching experience. Users can find new films they might not have otherwise known about by using the system to analyse user interests and offer pertinent recommendations. Additionally, the system can make it simpler for users to find and watch the films by offering a user-friendly webpage.

1.4 Methodology

The project's goal is to create a platform that can suggest movies to users, give in-depth descriptions of the movies they search for, and analyse user emotion in movie reviews. The time spent deciding which movie to view will be reduced thanks to the information offered. The major goal of developing a movie recommendation system is to give customers recommendations that are based on their favourite movies, not on popular movies or just ratings. As a result, the recommendation engine will generate highly customised recommendations, improving its accuracy. When choosing a movie, the user will benefit from the sentiment analysis and the additional details of the searched film. Due to the availability of all the necessary information on a single platform, the user won't need to search the internet in order to select a movie they like. The recommendation algorithm will give the user the top 10 films that are most similar to the searched movie, so they won't need to rely on their friends for movie recommendations.

1.4.1 Cosine similarity algorithm

Systems for suggesting films to viewers are created based on their tastes, past viewing habits, and behaviour. These programmes analyse user data and offer tailored recommendations using a variety of algorithms. One such tool used to determine how similar two films are is the cosine similarity algorithm.

In a multidimensional space, cosine similarity calculates the cosine of the angle between two vectors. Each movie is represented in movie recommendation systems as a vector in a high-dimensional space, with each dimension denoting a characteristic or attribute of the film. Examples of features of a film include its genre, director, cast, year of release, and audience reviews.

The cosine similarity algorithm computes the cosine of the angle between the feature vectors of two videos to determine how similar they are. A score of 1 indicates perfect resemblance between the two films, while a value of 0 indicates no similarity at all. The cosine of the angle is a measurement of this similarity. Because it is quick and effective, the cosine similarity algorithm is a preferred option in movie recommendation systems. Large datasets may be handled, and parallelization is simple. Furthermore, unlike other machine learning algorithms, it doesn't need to use a training dataset.

We must first develop a movie-feature matrix before we can incorporate the cosine similarity algorithm into a movie recommendation system. The features of each movie in the dataset are listed in the matrix, which is a table. A movie is represented by each row, and a feature by each column, in the matrix. For instance, the movie-feature matrix would be 100 x 10 if there were 100 films and 10 features. Once we have the movie-feature matrix, we can calculate the cosine similarity between two movies using the following formula:

cosine_similarity(A, B) = (A.B) / (||A||.||B||)

Where A and B are the feature vectors of the two movies, A.B is the dot product of the vectors, and ||A|| and ||B|| are the magnitudes of the vectors.

The cosine similarity calculation yields a value between 0 and 1, with 1 denoting complete identity between the two films and 0 denoting no similarity at all.

The steps listed below can be used to create movie suggestions using the cosine similarity algorithm:

1. Decide on the target movie for which you want recommendations.
2. Calculate the cosine similarity of the target movie to each other movie in the dataset.
3. Choose the top n films with the highest cosine similarity values to serve as the user's suggestions.

The programme determines the value of n, which can also be set based on system requirements or user preferences. The cosine similarity method can be utilised in movie recommendation systems for various tasks including movie clustering and genre classification in addition to producing movie suggestions. By grouping similar films together based on their similarities, we can give customers more specialised recommendations.

Overall, one of the most effective tools in movie recommendation systems is the cosine similarity algorithm. Without the need for a training dataset, it enables us to compare movies based just on their attributes. By using this algorithm, we can generate personalized recommendations for users and improve their movie-watching experience.

1.4.2 Python Libraries

1) Pandas

Pandas is the name of a Python module for manipulating and analysing data. It provides rapid, flexible, and expressive data structures that are designed to make working with "relational" or "labelled" data easy and intuitive. Pandas is essentially a tool for manipulating, purifying, and analysing data. It is the most used software library for modifying and analysing data. Pandas provides data structures for efficiently storing and managing large datasets as well as tools for data cleaning, merging, and analysis. Fast data loading, alignment, manipulation, and fusion are all made possible. The open-source Pandas package serves as the foundation for the Python programming language. It is commonly used in applications combining data science and machine learning.

Pandas are capable performing a wide range of jobs, including:

1. Pandas allows for the slicing of dataframes.
2. Pandas can be used to combine and merge data frames.
3. Join the columns of two data frames together.
4. To modify the data frame's index value.
5. In pandas NumPy, you can modify column heads.

2) NumPy

Big, multidimensional arrays and matrices are supported by the NumPy Python library for scientific computing, along with a multitude of high-level mathematical operations that may be carried out on these arrays. It was created in 2005 by combining the Python modules for Numeric and Num array. Due to the homogeneity of NumPy arrays, each element needs to be of the same data type. NumPy arrays are more efficient than Python lists for numerical operations since vectorized operations can be applied to entire arrays at once rather than to individual members. NumPy is a core library for many other scientific libraries, including SciPy, Matplotlib, and Pandas, and it is widely used in scientific computing and data analysis. One of the most well-known open-source Python modules, NumPy focuses on logical processing. It supports vast networks, a variety of data, and implicit number-related skills for quick calculations. The abbreviation "NumPy" stands for "Mathematical Python". It frequently serves as a straight polynomial math tool, an advanced general knowledge information storage unit, an irregular number generator, etc. The most helpful functions in NumPy include arcsin(), arccos(), tan(), and radians (). An N-layered display with lines and segments is described by a Python object called a NumPy Cluster. NumPy displays are frequently preferred for records in Python. It uses less memory since it operates more quickly and is simpler to use.

Images, sound waves, and other simple two-dimensional information streams can be regarded as N-layered variations of real characteristics for representation using the NumPy interface. To use this ML library, full-stack designers need a tremendous amount of knowledge.

3) Scikit Learn

Scikit-learn (sklearn), a well-known machine learning toolkit for Python, provides a number of techniques for classification, regression, clustering, and dimensionality reduction. It was built on top of NumPy, SciPy, and Matplotlib and is a well-liked tool for data scientists and machine learning to build predictive models.

Important traits of Scikit-learn include the following:

1. Effective and simple data analysis and mining tools.
2. Usable in a range of circumstances and accessible to everyone.
3. Made with the help of Matplotlib, SciPy, and NumPy.
4. Open source; BSD licence; used in trade.
5. Integrated methods and tools for dimensionality reduction, clustering, regression, and classification applications in machine learning.
6. Comprehensive documentation and team support.

Scikit-learn's model selection, evaluation, and tuning features make it simpler for users to improve their machine learning models. Additionally, it offers a number of datasets that may be used to test and exercise different machine learning methods.

4) SciPy

SciPy is a Python library for scientific computing that was constructed on top of the NumPy library. For tasks like optimisation, linear algebra, signal processing, image processing, and more, it offers a variety of methods and functions.

1. Its compatibility with NumPy, which facilitates its use with other scientific computing Python tools.
2. A complete collection of tools for scientific computing that performs tasks like signal processing and optimisation.
3. fast and efficient algorithms that are very effective.
4. Free to use and change software that is open source.
5. Has a sizable and vibrant user and developer community and is simple to use and well documented.

This library provides a large variety of solutions for addressing common issues in scientific computing, potentially saving developers a tons of time and effort. This is one of the main benefits of adopting SciPy. For instance, SciPy has functions for solving differential equations, which are frequently employed in simulations of engineering and physics problems. A number of scientific and engineering applications require the use of tools for numerical optimisation and integration.
Overall, SciPy is a strong and adaptable Python library that is suitable for a variety of scientific computing workloads.

6) Matplotlib

There are steveral tools which are available for creating data visualisations in the well-known Python charting library Matplotlib. The many different sorts of plots that may be created using the vast array of functions provided by Matplotlib include line plots, scatter plots, bar plots, and histograms. It also offers tools for adding titles, labels, and annotations to plots as well as for altering the style, colour, and other characteristics of the plot parts.Some of the key features of Matplotlib include:

1. One of Matplotlib's primary characteristics is its extensive support for many plot and visualisation kinds.
2. Simple interaction with SciPy and NumPy, two Python libraries for scientific computation.
3. Plot elements with a wide range of colour, line-style, and font-size options.
4. Assistance with interactive charting within Jupyter notebooks and other interactive settings.
5. Comprehensive documentation and a sizable, vibrant user and developer community.
6. From straightforward line plots to intricate, interactive visualisations, Matplotlib is frequently used in data science and scientific computing. Additionally, it is employed in media to produce data-driven stories and images as well as in education to teach data visualisation.

7) Seaborn

Seaborn is a Python data visualisation library that uses Matplotlib. It provides an advanced drawing tool for producing captivating and instructive statistical graphics. The Matplotlib-based Seaborn provides a more streamlined and aesthetically pleasing API for creating visualisations. Seaborn provides a wide range of tools for creating several plot formats, including scatter plots, line plots, bar plots, heatmaps, and more. Seaborn is extensively used in data science and machine learning applications and is particularly useful for tasks involving exploratory data analysis and data visualisation. The BSD licence is used for Seaborn's distribution.

1.4.3 Kaggle

Data scientists and machine learning enthusiasts can connect online at Kaggle. Users of Kaggle can work together, access and share datasets, use notebooks with GPU integration, and compete with other data scientists to solve data science problems. Kaggle provides a range of datasets and challenges for users to work on, as well as a community of data scientists and machine learning practitioners who can collaborate and share knowledge.

The following are some of Kaggle's standout features:

1. Two of Kaggle's main advantages are having access to a variety of datasets and working on real-world issues.
2. A team of machine learning and data science professionals with whom to collaborate and exchange expertise.
3. A website where users may create and share data science projects and visualisations.
4. Instruments for building and honing machine learning models, including support for well-liked frameworks like TensorFlow and PyTorch.
5. The opportunity to compete with other users to solve problems from the real world and win prizes.

Data scientists and machine learning practitioners frequently utilise Kaggle to hone their abilities, construct their portfolios, and work with other experts in the area. Additionally, it helps businesses and organisations find solutions to pressing issues and spot the best data scientists.

1.4.4 API

APIs are also becoming popular nowadays. These are a group of protocols, methods, and tools known as an application programming interface, or API for short, allow different software programmes to connect with one another. It can be thought of more simply as a messenger that allows two different systems to effortlessly share data and services.

Developers can interface with various systems and services through APIs without needing to understand the underlying programming. They operate by making certain protocols, libraries, or services accessible to other apps via the internet. As a result, programmers are now able to create new software applications by utilising the features of already installed systems or services.

1.4.5 TMDB API

The Movie Database (TMDb) is a well-known online resource for details about motion pictures, television shows, and other related media. It is frequently used by programmers to create applications that offer details like cast and crew, release dates, trailers, reviews, ratings, and other aspects of movies. TMDb's API, which gives programmers access to its huge data collection and lets them incorporate it into their applications, is one of its most crucial aspects.An API key, which can be obtained by creating a free account on the TMDb website, is required before utilising the TMDb API to fetch data. We can make HTTP queries to the TMDb API endpoints and retrieve data in JSON format after we have the API key.

1.4.6 Python flask

The lightweight web framework Flask is simple to understand and use. It offers a quick and flexible approach to create web apps, which makes it a great option for the front end of a movie recommendation system. A built-in development server provided by Flask makes it simple to test and troubleshoot your application locally.

You can build a recommendation system that gives customers tailored movie recommendations based on their likes and preferences by combining Flask with a machine learning model and an intuitive interface. Flask is a fantastic option for developing web apps of various types, including movie recommendation systems, because to its simplicity of use and adaptability.

1.5 Organization

Chapter 1: - This chapter provides a succinct overview of the project. The chapter provides a quick summary of the movie recommendation system as well as an introduction to the project. The project's overall problem statement and its aims are also discussed in this chapter. Along with providing information about the procedures in building the recommendation system, the chapter also offers a brief introduction to the approach utilised for the implementation of a movie recommendation system utilising several machine learning algorithms.

Chapter 2: - This chapter gives the knowledge about the previous work related to the various recommendation systems. This also provides the information related to the different machine learning techniques. I have mentioned various Journals and related papers which give information about the work done earlier. The chapter gave us the information about how the various researchers have tried to use the various methods in order to design recommendation systems. The techniques and the results for those techniques are mentioned in this chapter, and these help us to find the approach that we are going to use to create our model or project.

Chapter 3: - This chapter gave the information about technologies that I have used to make this project and their applications in various fields and how these technologies are implemented that are required to run the project and to create the project. I have also given a brief explanation of model design of the project that uses various algorithms. It also describes functional and non-functional requirements needed for the project. In the end, I have added the data flow and workflow of the whole project.

Chapter 4: - This chapter gives the information about that how the whole project work is done and at every stage how we have kept check on the work. It provides information about the work done at different levels and also give the results obtained at the different levels. It provides the information about the model that we have created using the different modules and libraries. It also consists of the results from the various performance measure that we have used in the project. It provides the information about the performance of the project The whole chapter is providing us the information about the performance our whole model or project.

Chapter 5: - This chapter consists of the whole conclusion of the work presented in this project report. It provides information about the whole phases of the project and it also mentions the future scope for the project. It also consists the information about the applications of the project and where the project can be utilized in order to make that sector more computerised. It gives the information that how we can improve the project and what we can do in future related to this project and how we can improve this project

CHAPTER 2: LITERATURE REVIEW

Over time, various recommendation systems had been developed by various techniques either by the content based, collaborative or by the hybrid based technique. These recommendation systems had been implemented by the use of algorithms of machine learning and big data. Movie recommendation systems have gained widespread popularity in recent years as they help users to find movies that they may like based on their past preferences. These systems use machine learning algorithms and various data sources such as movie metadata, user ratings, and reviews to provide personalized recommendations to users. In this literature review, I analyze and compare the movie recommendation systems proposed in various research papers. I have selected many research papers that use different machine learning techniques and data sources to develop movie recommendation systems.

Ahuja et al. (2019) [1] suggested a method for recommending films that groups users based on their movie ratings and recommends films to them based on the films highly rated by similar users. This system uses k-means clustering and the k-nearest neighbour algorithm. The outcomes demonstrated that the suggested approach outperformed the conventional collaborative filtering strategy. Airen and Agrawal (2022) [2] developed a movie recommendation system using k-nearest neighbors variants such as user-based, item-based, and hybrid approach. The outcomes demonstrated that the hybrid strategy performed better in terms of accuracy than other alternatives. A movie recommendation system that employs sentiment analysis to extract user opinions from reviews and recommend films based on user sentiment was put out by Chauhan et al. (2021) [3]. The results showed that the suggested solution outperformed more traditional collaborative filtering techniques. An unsupervised machine learning-based movie recommendation system was developed by Putri et al. (2020) [4] that uses clustering techniques like k-means and hierarchical clustering to categorise movies according to how similar they are. The results showed that the suggested method performed more effectively than the traditional collaborative filtering approach. A movie recommendation system that proposes movies to viewers using machine learning techniques like decision trees, random forests, and gradient boosting was proposed by Furtado and Singh (2020) [5]. The results showed that the gradient boosting technique outperformed other algorithms in terms of accuracy. Goyani and Chaurasiya (2020) [6] examined the various techniques used in a review of movie recommendation systems, including collaborative filtering, content-based filtering, and hybrid approaches. The outcomes of the combined technology tell us that they have a good functionality than other algorithms. Khatter et al. generated a model using two algorithms [7]. The outcomes of the combined technology tell us that they have a good functionality than other algorithms. In 2021[8], Mehta and Gupta generated a model using two algorithms. The outcomes of the combined technology tell us that they have a good accuracy than other algorithms. Reddy et al. (2019) [9] created an algorithm that utilizes the data from the training data. The outcomes shows that the new algorithm give good results as compared to old algorithms. A movie recommendation system using cosine similarity and the k-nearest neighbour technique was put out by Singh et al. (2020) [10]. The outcomes demonstrated that the suggested strategy worked better than conventional collaborative filtering techniques. Smitha et al.'s evaluation of movie recommendation systems from 2021 [11] evaluated the various methods utilised, including collaborative filtering, content-based filtering, and hybrid approaches. The findings demonstrated that hybrid techniques perform better in terms of accuracy than other approaches. A hybrid recommendation system that incorporates collaborative filtering and deep learning methods for movie suggestion is proposed by Kaur, P., Singh, N., & Singh, K. (2021) [12]. The system initially creates user-item ratings using collaborative filtering, and after that it employs a deep neural network to discover the latent properties of users and films. The experimental results demonstrate that the suggested system outperforms existing cutting-edge recommendation systems as well as conventional collaborative filtering techniques. An improved K-nearest neighbour (KNN) algorithm for movie recommendation is put out by Singh, R. K., & Tripathi S. K. (2021) [13]. The suggested algorithm makes use of a modified distance calculation that accounts for user rating patterns and item popularity. The experimental findings demonstrate that the suggested algorithm performs better in terms of accuracy and diversity than conventional KNN and other cutting-edge recommendation algorithms. A movie recommendation system based on semantic feature extraction and clustering is suggested by Ahmad, W., Javed, M. Y., Imran, & Hussain (2021) [14]. The system first uses natural language processing methods to extract semantic elements from the movie plot summaries before applying clustering algorithms to group related movies together. The experimental results demonstrate that the suggested method outperforms other cutting-edge recommendation systems as well as conventional content-based recommendation systems. A movie recommendation system built on a convolutional neural network (CNN) with feature combination is suggested by Lee, J., Lee, Y., Lee, J., & Kwon, O. (2021) [15] proposes a movie recommendation system based on a convolutional neural network (CNN) with feature combination. The system first extracts visual features from movie posters and text features from plot summaries and then combines them using a CNN to generate recommendations. The experimental results show that the proposed system outperforms traditional content-based and collaborative filtering recommendation systems and achieves state-of-the-art performance. Singh, S., & Singh, S. (2021) [16] proposes an enhanced movie recommendation system that combines fuzzy clustering and cosine similarity. The system first uses fuzzy clustering to group similar movies together and then uses cosine similarity to generate recommendations. The experimental results show that the proposed system outperforms traditional content-based and collaborative filtering recommendation systems and achieves state-of-the-art performance. .Lee, J., & Kwon, O. (2021) [15]. The algorithm initially collects text and visual elements from plot summaries and movie posters, combining them with a CNN to produce recommendations. The results of the experiments show that the proposed system works at a cutting-edge level and outperforms traditional content-based and collaborative filtering recommendation systems. Using fuzzy clustering and cosine similarity, Singh, S., & Singh, S. (2021) [16] propose an enhanced movie selection algorithm. Before using cosine similarity to offer choices, the system groups related films together using fuzzy clustering. The trials' findings demonstrate that the suggested system performs better than both conventional content-based and collaborative filtering recommendation systems and operates at a cutting-edge level. (2017) [17] A hybrid deep learning-based recommendation system is put forth by Li, L., Li, J., and Zhang. The technique combines a deep neural network with a collaborative filtering process to improve the accuracy of movie selections. The proposed technique is evaluated on the MovieLens dataset, and the findings show that it outperforms current state-of-the-art recommendation algorithms. Yu, F., and Yang, J. (2019) [18] offer a novel deep learning model for movie recommendation that combines content-based and collaborative filtering methods. The proposed approach is based on a convolutional neural network (CNN) that has been trained on movie trailers and reviews. The experimental findings show that the suggested model outperforms other state-of-the-art recommendation algorithms on the MovieLens dataset. Zhang, X., Zhao, and Leung (2020) [19] propose a deep learning-based hybrid autoencoder-based technique for group movie selection. The algorithm is designed to handle the sparsity and cold-start problems of recommendation systems. On the MovieLens dataset, the proposed algorithm is tested, and the results demonstrate that it performs better than other cutting-edge recommendation algorithms. A hybrid deep learning model for movie selection that integrates both content-based and collaborative filtering approaches is presented by Zhang, X., and Leung, H. F. (2020) [20]. Long short-term memory (LSTM) network and convolutional neural network make up the suggested model. On the MovieLens dataset, the experimental results demonstrate that the suggested model performs better than other cutting-edge recommendation algorithms. Using deep learning, Liu, X., Wang, W., Wu, & Xu (2021) [21] suggests a better movie recommendation method. A deep learning module plus a collaborative filtering module make up the suggested algorithm. A convolutional neural network (CNN) that has been trained using movie trailers and reviews serves as the foundation of the deep learning module. On the MovieLens dataset, the experimental results demonstrate that the suggested algorithm performs better than other cutting-edge recommendation algorithms.

CHAPTER 3: SYSTEM DEVELOPMENT

3.1 Technologies Implementation

Here I have discussed about the tools and technologies I used in detail:

- Python - Used for preprocessing and creating models.

Python has gained a lot of popularity over the years because of how easily it can be modified and how straightforward it is. It is frequently used in a variety of various fields, including as artificial intelligence, machine learning, data analysis, and web development.
One of the most widely used programming languages in the machine learning field is Python. Python is frequently utilised for prepping data and building machine learning models. Utilising statistical models and techniques, machine learning enables computer systems to learn from data and draw conclusions without explicit programming.
Many Python modules and tools make it simple to build machine learning models and prepare the data for them. A few examples of libraries that offer a variety of functions and techniques for cleaning, altering, and manipulating data are NumPy, Pandas, and Scikit-Learn.
Additionally, Python provides a wide range of visualisation tools, which are crucial for deciphering and understanding data. Libraries like Matplotlib, Seaborn, and Plotly provide a variety of visualisation features for the production of graphs, charts, and other visual aids.

- HTML5 - Used for creating user-friendly web pages.

HTML5 is a markup language that is frequently used to create interesting and user-friendly web pages. Thanks to its broad range of features and capabilities, developers may construct logical, dynamic web pages that are easy to navigate. Another important feature of HTML5 is its support for responsive design, which enables web pages to quickly adapt to various screen sizes and orientations. As more people access the internet via mobile devices, HTML5 makes it easier to construct responsive web pages. To name a few of the tools HTML5 enables developers to create user-friendly websites include forms, tables, lists, and media elements. These elements help order the content and structure of the web page, creating a more organised and user-friendly surfing experience.

- CSS3 - Used for styling the Html page.

It is feasible to create web pages that are both aesthetically beautiful and user-friendly with the aid of HTML5 and CSS3. Given that it allows developers access to a wide number of tools and features for designing the structure, typography, and colour schemes of web pages, it is an essential tool for creating modern and professional-looking websites.

HTML page style is one of CSS3's main applications. A wide variety of visual effects, including as backgrounds, borders, shadows, gradients, and animations, may be produced by developers using CSS3. Additionally, CSS3's improved typographic features let developers change the font family, size, weight, and spacing of text on websites.

Another crucial CSS3 feature is the capacity to create responsive web design. As more people access the internet using a variety of devices, websites must be designed to adapt to different screen sizes and orientations. Thanks to CSS3, which makes it easier to create flexible and adaptable layouts that dynamically adjust to multiple screen sizes, a better user experience is created for all users.

The flexibility of CSS3 to create reusable and modular styles makes it easier to manage and change the styling of web pages. Using classes and IDs, developers can apply the same styles to a variety of web page elements, resulting in a cohesive and consistent design.

- JavaScript - Used for adding dynamic features and special effects on the webpage.

Web pages frequently have dynamic elements and special effects added using the well-liked programming language JavaScript. The fact that it is a client-side language, which means that it runs in the user's web browser, enables programmers to construct dynamic web pages that react to user input.

JavaScript is frequently used to build dynamic web pages that change content without requiring a page refresh. This language used to dynamically load new content as a user scrolls across a website or to rapidly update specific pages,. Additionally, JavaScript gives programmers a large selection of tools for adding animations and special effects to web pages. These can consist of image sliders, scrolling animations, hover effects, and much more. JavaScript enables developers to create interactive and engaging web pages that keep users interested in the content.

The development of interactive forms and user interfaces is a significant additional usage of JavaScript. JavaScript, for instance, can be used to instantly check user input on forms and give users feedback as they enter information. JavaScript can also be used to develop unique user interfaces, like sliders, dropdown menus, and other interactive features, which facilitate user interaction and navigation of web pages.

Python is mostly used in the suggested solution to work with data sets and apply various machine learning algorithms to produce the desired results. For a better user experience, dynamic web pages and an intuitive GUI are also made using AJAX, HTML, and JSON. Six steps primarily make up the implementation process. These are what they are:

Step 1: Finding and loading appropriate data is step one. Shortlisted suitable data sets are retrieved from Kaggle. The data for the last three years is taken from Wikipedia to keep the data current. IMDb (Internet Movie Database) retrieves movie reviews in order to conduct sentiment analysis. The TMDB (The Movie Database) API is also used to retrieve additional information, cast photos, and movie posters.

Step 2: Cleaning the data Both the data sets and the Wikipedia data were obtained from Kaggle, and both were cleaned in VScode. The primary CSV file, which will be used whenever the data needs to be accessed, is then loaded with the cleaned data.

Step 3: Creating an API key in step three API is an abbreviation for "Application Programming Interface." A software bridge called an API enables communication between two programmes. In other words, an API is a messenger that transmits your request to the service provider you are using and returns the result. The provider in this instance is TMDB. The machine can get the data it requires from TMDB's enormous library of movie data. After creating an account on TMDB, an API key must be generated in order to use the TMDB API.

Step 4: Conducted Sentiment Analysis Python is used to import the NLTK (Natural Language Toolkit) library and conduct different operations on the reviews data. Imported NLTK corpus is used to examine various Natural Language data sets. Tokenizing the data, learning the vocabulary, and inverting document frequency weightings are all functions of the Tfidf vectorizer. The categorization and analysis of the data, which is subsequently divided into training and testing data in a 4:1 ratio, are performed using the multinomial Naive Bayes algorithm.

Step 5: Producing Web page files The GUI must be effective for the system to be practical and simple to use. This would make it easier for the user to interact with the software. The user is sent to two major HTML sites, one for searching for movies and the other for seeing movie details, sentiment analysis of movie reviews, and top ten movie recommendations. JSON is a data format that is used to send data to the server, and AJAX is used to retrieve data from the server.

Step 6: Using Machine Learning Algorithms for Recommendation On the main data set, the count vectorizer function is used to create a count matrix. Utilising the frequency (count) of each word that appears in the data set, the Count Vectorizer function converts the data into a vector. The cosine istance is then calculated using Cosine Similarity on those vectors in order to recommend the top ten films that are most similar to the searched movie.

3.2 Functional Requirements

A recommendation system is a paradigm for information filtering that aims to identify user preferences and make recommendations in accordance with those choices. The problem of wasting a lot of time looking for our favourite films to watch is alleviated by movie recommendation systems, which assist us in searching for our preferred films among all the available sources. Therefore, it is crucial that the system used to recommend films be very accurate and trustworthy so that it can give us recommendations for films that are either exactly the same as our preferences or most closely match them.

Table 3.2.1 Movie recommendation based on different ages

User	Recommendations
Student	Get movie recommendations related to adventures.
	Get the animated movie recommendations.
Adult	Get the movie recommendations related to erotic.
	Get movie recommendations related to comedy.
Old	Get the religious movie recommendation.
	Get the movie recommendations related to biopics.

3.3 Non-Functional Requirements

Performance of our website depends on the time taken by the recommendation engine to get the results with the movie details, cast details and the recommendations/suggestions of similar types of movies.

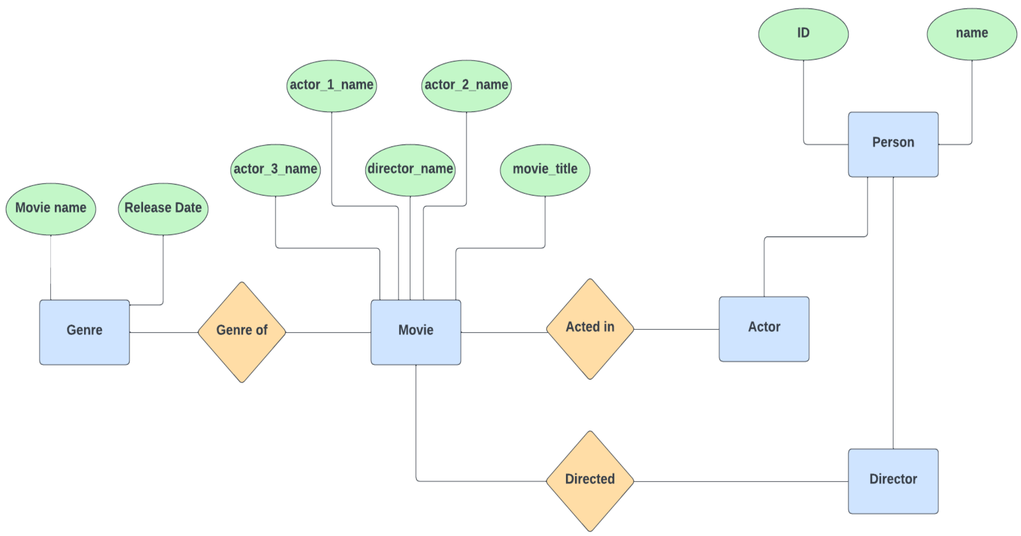

figure 3.3.1: E-R Diagram of the movie recommendation system

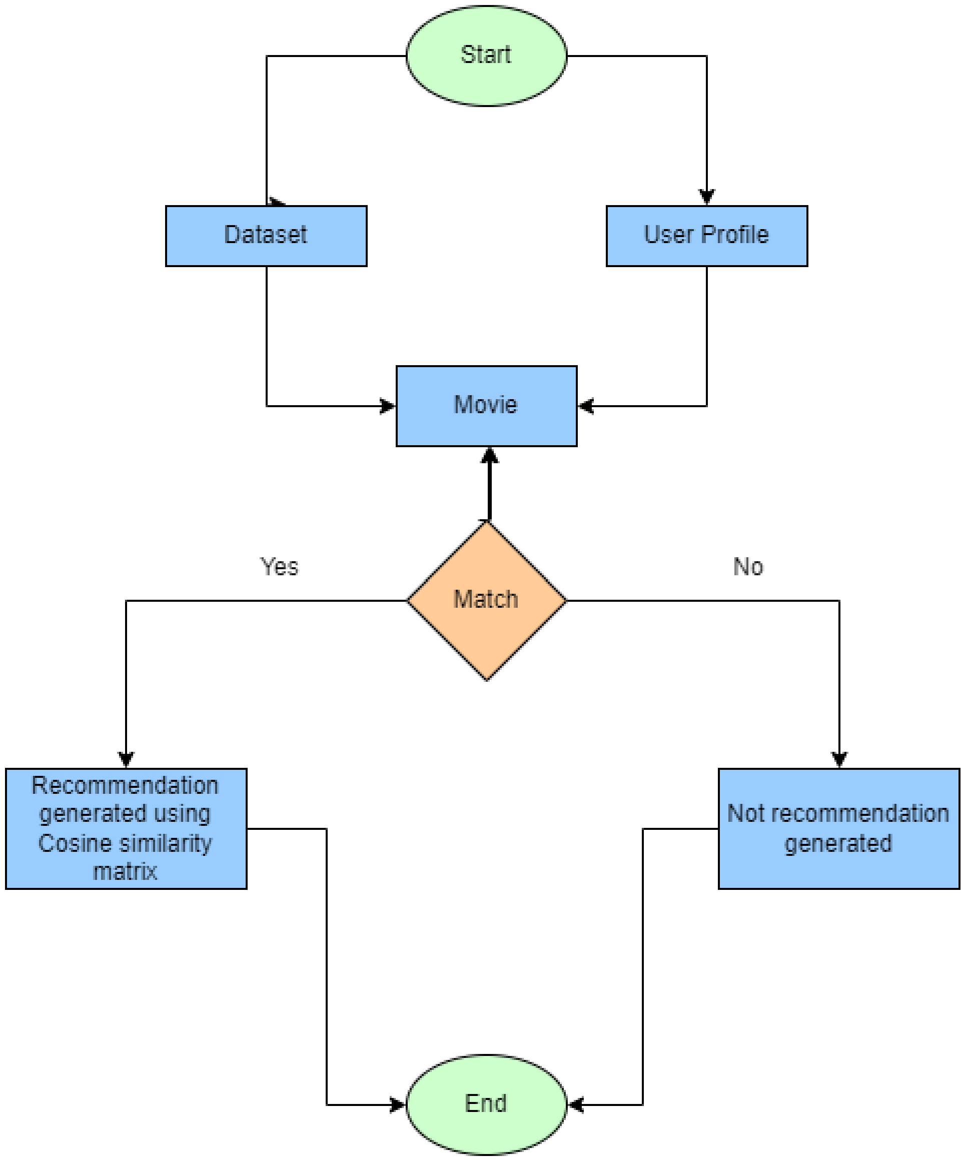

Figure 3.3.2: Work flow of the project

CHAPTER 4: PERFORMANCE ANALYSIS

4.1 Data Set Used in the Minor Project

In this project I have used various datasets which includes:

a. IMDB 5000 Movies Datasets:

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes	actor_2_name	actor_1_facebook_likes	gross
0	Color	James Cameron	723.0	178.0	0.0	855.0	Joel David Moore	1000.0	760505847.0
1	Color	Gore Verbinski	302.0	169.0	563.0	1000.0	Orlando Bloom	40000.0	309404152.0
2	Color	Sam Mendes	602.0	148.0	0.0	161.0	Rory Kinnear	11000.0	200074175.0
3	Color	Christopher Nolan	813.0	164.0	22000.0	23000.0	Christian Bale	27000.0	448130642.0
4	NaN	Doug Walker	NaN	NaN	131.0	NaN	Rob Walker	131.0	NaN
5	Color	Andrew Stanton	462.0	132.0	475.0	530.0	Samantha Morton	640.0	73058679.0
6	Color	Sam Raimi	392.0	156.0	0.0	4000.0	James Franco	24000.0	336530303.0
7	Color	Nathan Greno	324.0	100.0	15.0	284.0	Donna Murphy	799.0	200807262.0
8	Color	Joss Whedon	635.0	141.0	0.0	19000.0	Robert Downey Jr.	26000.0	458991599.0
9	Color	David Yates	375.0	153.0	282.0	10000.0	Daniel Radcliffe	25000.0	301956980.0

10 rows × 28 columns

Figure 4.1.1: movie_metadata.csv

b.The Movies Dataset:

It contains the following files:

i. <u>Movies_metadata.csv:</u> The primary Movies Metadata file, which is part of the Full MovieLens collection, includes details on 45,000 films.

	genres	id	title	year
26560	[{'id': 12, 'name': 'Adventure'}, {'id': 28, '...	166426	Pirates of the Caribbean: Dead Men Tell No Tales	2017.0
26561	[{'id': 28, 'name': 'Action'}, {'id': 12, 'nam...	141052	Justice League	2017.0
26565	[{'id': 28, 'name': 'Action'}, {'id': 12, 'nam...	284053	Thor: Ragnarok	2017.0
26566	[{'id': 28, 'name': 'Action'}, {'id': 12, 'nam...	283995	Guardians of the Galaxy Vol. 2	2017.0
30536	[{'id': 14, 'name': 'Fantasy'}, {'id': 28, 'na...	245842	The King's Daughter	2017.0
...	...	...	...	...
45398	[{'id': 10749, 'name': 'Romance'}, {'id': 35, ...	468707	Thick Lashes of Lauri Mäntyvaara	2017.0
45417	[{'id': 80, 'name': 'Crime'}, {'id': 35, 'name...	461297	Cop and a Half: New Recruit	2017.0
45437	[{'id': 10751, 'name': 'Family'}, {'id': 16, '...	455661	In a Heartbeat	2017.0
45453	[{'id': 80, 'name': 'Crime'}, {'id': 18, 'name...	404604	Mom	2017.0
45465	[]	461257	Queerama	2017.0

532 rows × 4 columns

Figure 4.1.2: movies_metadata.csv

ii. credits.csv: It contains details about the cast and crew for all of our films. It is accessible as a JSON Object that has been stringified.

	cast	crew	id
0	[{'cast_id': 14, 'character': 'Woody (voice)',...	[{'credit_id': '52fe4284c3a36847f8024f49', 'de...	862
1	[{'cast_id': 1, 'character': 'Alan Parrish', '...	[{'credit_id': '52fe44bfc3a36847f80a7cd1', 'de...	8844
2	[{'cast_id': 2, 'character': 'Max Goldman', 'c...	[{'credit_id': '52fe466a9251416c75077a89', 'de...	15602
3	[{'cast_id': 1, 'character': "Savannah 'Vannah...	[{'credit_id': '52fe44779251416c91011acb', 'de...	31357
4	[{'cast_id': 1, 'character': 'George Banks', '...	[{'credit_id': '52fe44959251416c75039ed7', 'de...	11862
...	...	...	...
45471	[{'cast_id': 0, 'character': '', 'credit_id': ...	[{'credit_id': '5894a97d925141426c00818c', 'de...	439050
45472	[{'cast_id': 1002, 'character': 'Sister Angela...	[{'credit_id': '52fe4af1c3a36847f81e9b15', 'de...	111109
45473	[{'cast_id': 6, 'character': 'Emily Shaw', 'cr...	[{'credit_id': '52fe4776c3a368484e0c8387', 'de...	67758
45474	[{'cast_id': 2, 'character': '', 'credit_id': ...	[{'credit_id': '533bccebc3a36844cf0011a7', 'de...	227506
45475	[]	[{'credit_id': '593e676c92514105b702e68e', 'de...	461257

45476 rows × 3 columns

Figure 4.1.3: credits.csv

c. We are
also extracting features of movies of various years from Wikipedia.

4.2 Data Set Features

Since, we are having various datasets with many different features, thus we have extracted various features from all datasets and have appended them and have created a final dataset. To do this , we have firstly selected a few features on which my recommendation will be done. These features include 'director_name', 'actor_1_name', 'actor_2_name', 'actor_3_name', 'genres', 'movie_title'.

```
data = data.loc[:,['director_name', 'actor_1_name', 'actor_2_name',
'actor_3_name', 'genres', 'movie_title']]
```

The above line of code extracted the required features from movie_metadat.csv (IMDB 5000 Movies Dataset). In this dataset we got data of movies till the year 2016. So, we will extract data of movies of next years from other datasets that we have.

	director_name	actor_1_name	actor_2_name	actor_3_name	genres	movie_title
0	James Cameron	CCH Pounder	Joel David Moore	Wes Studi	Action Adventure Fantasy Sci-Fi	avatar
1	Gore Verbinski	Johnny Depp	Orlando Bloom	Jack Davenport	Action Adventure Fantasy	pirates of the caribbean: at world's end
2	Sam Mendes	Christoph Waltz	Rory Kinnear	Stephanie Sigman	Action Adventure Thriller	spectre
3	Christopher Nolan	Tom Hardy	Christian Bale	Joseph Gordon-Levitt	Action Thriller	the dark knight rises
4	Doug Walker	Doug Walker	Rob Walker	unknown	Documentary	star wars: episode vii - the force awakens ...
5	Andrew Stanton	Daryl Sabara	Samantha Morton	Polly Walker	Action Adventure Sci-Fi	john carter
6	Sam Raimi	J.K. Simmons	James Franco	Kirsten Dunst	Action Adventure Romance	spider-man 3
7	Nathan Greno	Brad Garrett	Donna Murphy	M.C. Gainey	Adventure Animation Comedy Family Fantasy Musi...	tangled
8	Joss Whedon	Chris Hemsworth	Robert Downey Jr.	Scarlett Johansson	Action Adventure Sci-Fi	avengers: age of ultron
9	David Yates	Alan Rickman	Daniel Radcliffe	Rupert Grint	Adventure Family Fantasy Mystery	harry potter and the half-blood prince

Figure 4.2.1: movie_metadata.csv after selecting required features

After this, we are going to use the movies metadata.csv (The Movies Dataset) to get movies data of the year 2017.

	genres	id	title	year
26560	[{'id': 12, 'name': 'Adventure'}, {'id': 28, '...	166426	Pirates of the Caribbean: Dead Men Tell No Tales	2017.0
26561	[{'id': 28, 'name': 'Action'}, {'id': 12, 'nam...	141052	Justice League	2017.0
26565	[{'id': 28, 'name': 'Action'}, {'id': 12, 'nam...	284053	Thor: Ragnarok	2017.0
26566	[{'id': 28, 'name': 'Action'}, {'id': 12, 'nam...	283995	Guardians of the Galaxy Vol. 2	2017.0
30536	[{'id': 14, 'name': 'Fantasy'}, {'id': 28, 'na...	245842	The King's Daughter	2017.0
...	...	...	...	...
45398	[{'id': 10749, 'name': 'Romance'}, {'id': 35, ...	468707	Thick Lashes of Lauri Mäntyvaara	2017.0
45417	[{'id': 80, 'name': 'Crime'}, {'id': 35, 'name...	461297	Cop and a Half: New Recruit	2017.0
45437	[{'id': 10751, 'name': 'Family'}, {'id': 16, '...	455661	In a Heartbeat	2017.0
45453	[{'id': 80, 'name': 'Crime'}, {'id': 18, 'name...	404604	Mom	2017.0
45465	[]	461257	Queerama	2017.0

532 rows × 4 columns

Figure 4.2.2: Data of movies of year 2017 collected from movies_metadata.csv

Now we merged this data with credits.csv.

	genres	id	title	year	cast	crew
0	[{'id': 12, 'name': 'Adventure'}, {'id': 28, 'name': 'Action'}, {'id': ...	166426	Pirates of the Caribbean: Dead Men Tell No Tales	2017.0	[{'cast_id': 1, 'character': 'Captain Jack Sparrow', 'credit_id': '52fe...	[{'credit_id': '52fe4c9cc3a36847f8236a65', 'department': 'Production', ...
1	[{'id': 28, 'name': 'Action'}, {'id': 12, 'name': 'Adventure'}, {'id': ...	141052	Justice League	2017.0	[{'cast_id': 2, 'character': 'Bruce Wayne / Batman', 'credit_id': '535e...	[{'credit_id': '55ef66dbc3a3686f1700a52d', 'department': 'Production', ...
2	[{'id': 28, 'name': 'Action'}, {'id': 12, 'name': 'Adventure'}, {'id': ...	284053	Thor: Ragnarok	2017.0	[{'cast_id': 0, 'character': 'Thor Odinson', 'credit_id': '545d46a80e0a...	[{'credit_id': '56a93fa4c3a36872db001e7a', 'department': 'Writing', 'ge...
3	[{'id': 28, 'name': 'Action'}, {'id': 12, 'name': 'Adventure'}, {'id': ...	283995	Guardians of the Galaxy Vol. 2	2017.0	[{'cast_id': 3, 'character': 'Peter Quill / Star-Lord', 'credit_id': '5...	[{'credit_id': '59171547925141583c0315a6', 'department': 'Sound', 'gend...
4	[{'id': 14, 'name': 'Fantasy'}, {'id': 28, 'name': 'Action'}, {'id': 12...	245842	The King's Daughter	2017.0	[{'cast_id': 0, 'character': 'King Louis XIV', 'credit_id': '5431dd580e...	[{'credit_id': '5431de49c3a36825d300007e', 'department': 'Directing', '...
...	...	...	...	...	...	...
526	[{'id': 10749, 'name': 'Romance'}, {'id': 35, 'name': 'Comedy'}]	468707	Thick Lashes of Lauri Mäntyvaara	2017.0	[{'cast_id': 0, 'character': 'Satu', 'credit_id': '597e2086c3a368544001...	[{'credit_id': '597e22f69251415d7801c74a', 'department': 'Directing', '...
527	[{'id': 80, 'name': 'Crime'}, {'id': 35, 'name': 'Comedy'}, {'id': 28, ...	461297	Cop and a Half: New Recruit	2017.0	[{'cast_id': 0, 'character': 'Detective Simmons', 'credit_id': '593ba04...	[{'credit_id': '593ba0c29251410593009be3', 'department': 'Writing', 'ge...
528	[{'id': 10751, 'name': 'Family'}, {'id': 16, 'name': 'Animation'}, {'id...	455661	In a Heartbeat	2017.0	[]	[{'credit_id': '5981a15c92514151e0011b51', 'department': 'Sound', 'gend...
529	[{'id': 80, 'name': 'Crime'}, {'id': 18, 'name': 'Drama'}, {'id': 53, '...	404604	Mom	2017.0	[{'cast_id': 1, 'character': 'Devki Sabarwal', 'credit_id': '577809adc3...	[{'credit_id': '58ee55bbc3a3683df500bd0f', 'department': 'Sound', 'gend...
530	[]	461257	Queerama	2017.0	[]	[{'credit_id': '593e676c92514105b702e68e', 'department': 'Directing', '...

531 rows × 6 columns

Figure 4.2.3: Data after merging credits.csv with new_meta

Since, the features like ‘genres’, ‘cast’ and ‘crew’ contain data as JSON objects, we need to evaluate them using the “ast” module.

Now for creating a ‘genre_list’, we defined the needed function.

Finally we get the ‘genre_list’ as follows:

```
0      Adventure Action Fantasy Comedy
1      Action Adventure Fantasy Sci-Fi
2      Action Adventure Fantasy Sci-Fi
3       Action Adventure Comedy Sci-Fi
4             Fantasy Action Adventure
                    ...
526                     Romance Comedy
527       Crime Comedy Action Family
528    Family Animation Romance Comedy
529              Crime Drama Thriller
530                                NaN
Name: genres_list, Length: 531, dtype: object
```

Figure 4.2.4: ‘genre_list’ evaluated from the JSON object

Similarly, we evaluate a list of data for ‘actor_1_name’, ‘actor_2_name’, ‘actor_3_name’ and ‘director_name’.

Then we finally extract all these features with the same feature names as in the previous csv file and then merge them to a new file.

	director_name	actor_1_name	actor_2_name	actor_3_name	genres	movie_title	comb
0	James Cameron	CCH Pounder	Joel David Moore	Wes Studi	Action Adventure Fantasy Sci-Fi	avat	CCH Pounder Joel David Moore Wes Studi James Cameron Action Adventure F...
1	Gore Verbinski	Johnny Depp	Orlando Bloom	Jack Davenport	Action Adventure Fantasy	pirates of the caribbean: at world's e	Johnny Depp Orlando Bloom Jack Davenport Gore Verbinski Action Adventur...
2	Sam Mendes	Christoph Waltz	Rory Kinnear	Stephanie Sigman	Action Adventure Thriller	spect	Christoph Waltz Rory Kinnear Stephanie Sigman Sam Mendes Action Adventu...
3	Christopher Nolan	Tom Hardy	Christian Bale	Joseph Gordon-Levitt	Action Thriller	the dark knight ris	Tom Hardy Christian Bale Joseph Gordon-Levitt Christopher Nolan Action ...
4	Doug Walker	Doug Walker	Rob Walker	unknown	Documentary	star wars: episode vii - the force awakens	Doug Walker Rob Walker unknown Doug Walker Documentary
...	...	...	...	...	...	...	...
524	Jim Strouse	Jessica Williams	Chris O'Dowd	Keith Stanfield	Romance Comedy	the incredible jessica james	Jessica Williams Chris O'Dowd Keith Stanfield Jim Strouse Romance Comedy
525	Farhad Mann	Adelaide Kane	Benjamin Hollingsworth	Jean Louisa Kelly	Romance	can't buy my love	Adelaide Kane Benjamin Hollingsworth Jean Louisa Kelly Farhad Mann Romance
526	Hannaleena Hauru	Inka Haapamäki	Rosa Honkonen	Tiitus Rantala	Romance Comedy	thick lashes of lauri mäntyvaara	Inka Haapamäki Rosa Honkonen Tiitus Rantala Hannaleena Hauru Romance Co...
527	Jonathan A. Rosenbaum	Lou Diamond Phillips	Wallace Shawn	Gina Holden	Crime Comedy Action Family	cop and a half: new recruit	Lou Diamond Phillips Wallace Shawn Gina Holden Jonathan A. Rosenbaum Cr...
529	Ravi Udyawar	Sridevi Kapoor	Sajal Ali	Akshaye Khanna	Crime Drama Thriller	mom	Sridevi Kapoor Sajal Ali Akshaye Khanna Ravi Udyawar Crime Drama Thriller

5271 rows × 7 columns

Figure 4.2.5: 'new_data.csv'

Now, since we –don't have data for movies after 2017. So, we will be extracting it from wikipedia and for this we need to have libraries like 'bs4' and 'html5lib'. After we are done with the importing of these libraries then we run the script.

	Opening	Opening.1	Title	Production company	Cast and crew	.mw-parser-output .tooltip-dotted{border-bottom:1px dotted;cursor:help}Ref.	Ref.
0	JANUARY	5	Insidious: The Last Key	Universal Pictures / Blumhouse Productions / S...	Adam Robitel (director); Leigh Whannell (scree...	[2]	NaN
1	JANUARY	5	The Strange Ones	Vertical Entertainment	Lauren Wolkstein (director); Christopher Radcl...	[3]	NaN
2	JANUARY	5	Stratton	Momentum Pictures	Simon West (director); Duncan Falconer, Warren...	[4]	NaN
3	JANUARY	10	Sweet Country	Samuel Goldwyn Films	Warwick Thornton (director); David Tranter, St...	[5]	NaN
4	JANUARY	12	The Commuter	Lionsgate / StudioCanal / The Picture Company	Jaume Collet-Serra (director); Byron Willinger...	[6]	NaN
...	...	...	...	...	...	...	...
267	DECEMBER	25	Holmes & Watson	Columbia Pictures / Gary Sanchez Productions	Etan Cohen (director/screenplay); Will Ferrell...	NaN	[164]
268	DECEMBER	25	Vice	Annapurna Pictures / Plan B Entertainment	Adam McKay (director/screenplay); Christian Ba...	NaN	[137]
269	DECEMBER	25	On the Basis of Sex	Focus Features	Mimi Leder (director); Daniel Stiepleman (scre...	NaN	[228]
270	DECEMBER	25	Destroyer	Annapurna Pictures	Karyn Kusama (director); Phil Hay, Matt Manfre...	NaN	[260]
271	DECEMBER	28	Black Mirror: Bandersnatch	Netflix	David Slade (director); Charlie Brooker (scree...	NaN	[261]

272 rows × 7 columns

Figure 4.2.6: Dataframe we got from wikipedia for movies in 2018

As we can see that we don't have all the required features for our recommendation system, so we will be getting these features like genre from an api named “TMDB”. To connect to this api we first require to create an account on https://www.themoviedb.org/ and then get the api key from there to be used. We also need to have certain packages like ‘tmdbv3api’, ‘requests’ installed and imported for further preprocessing. Then we run the script to append the feature named ‘genres’ to the dataframe.

After running the script we get the genres list in the dataframe. After this we remove other features which are not useful for our analysis.

	Title	Cast and crew	genres
0	Insidious: The Last Key	Adam Robitel (director); Leigh Whannell (scree...	Horror Mystery Thriller
1	The Strange Ones	Lauren Wolkstein (director); Christopher Radcl...	Thriller Drama
2	Stratton	Simon West (director); Duncan Falconer, Warren...	Action Thriller
3	Sweet Country	Warwick Thornton (director); David Tranter, St...	Drama History Western
4	The Commuter	Jaume Collet-Serra (director); Byron Willinger...	Action Thriller Mystery
...	...	...	...
267	Holmes & Watson	Etan Cohen (director/screenplay); Will Ferrell...	Mystery Adventure Comedy Crime
268	Vice	Adam McKay (director/screenplay); Christian Ba...	Thriller Science Fiction Action Adventure
269	On the Basis of Sex	Mimi Leder (director); Daniel Stiepleman (scre...	Drama History
270	Destroyer	Karyn Kusama (director); Phil Hay, Matt Manfre...	Thriller Crime Drama Action
271	Black Mirror: Bandersnatch	David Slade (director); Charlie Brooker (scree...	Science Fiction Mystery Drama Thriller TV Movie

272 rows × 3 columns

Figure 4.2.7: Dataframe after keeping required features.

Now we extract features like ‘actor_1_name’, ‘actor_2_name’, ‘actor_3_name’ and ‘director_name’ from the “Cast and crew” column. And after all the final preprocessing, we get the data for movies in the year 2018.

	director_name	actor_1_name	actor_2_name	actor_3_name	genres	movie_title	comb
0	Adam Robitel	Lin Shaye	Angus Sampson	Leigh Whannell	Horror Mystery Thriller	insidious: the last key	Lin Shaye Angus Sampson Leigh Whannell Adam Ro...
1	Lauren Wolkstein	Alex Pettyfer	James Freedson-Jackson	Emily Althaus	Thriller Drama	the strange ones	Alex Pettyfer James Freedson-Jackson Emily Alt...
2	Simon West	Dominic Cooper	Austin Stowell	Gemma Chan	Action Thriller	stratton	Dominic Cooper Austin Stowell Gemma Chan Simon...
3	Warwick Thornton	Bryan Brown	Sam Neill	unknown	Drama History Western	sweet country	Bryan Brown Sam Neill unknown Warwick Thornton...
4	Jaume Collet-Serra	Liam Neeson	Vera Farmiga	Patrick Wilson	Action Thriller Mystery	the commuter	Liam Neeson Vera Farmiga Patrick Wilson Jaume ...
...	...	...	...	...	...	...	...
267	Etan Cohen	Will Ferrell	John C. Reilly	Rebecca Hall	Mystery Adventure Comedy Crime	holmes & watson	Will Ferrell John C. Reilly Rebecca Hall Etan ...
268	Adam McKay	Christian Bale	Amy Adams	Steve Carell	Thriller Science Fiction Action Adventure	vice	Christian Bale Amy Adams Steve Carell Adam McK...
269	Mimi Leder	Felicity Jones	Armie Hammer	Justin Theroux	Drama History	on the basis of sex	Felicity Jones Armie Hammer Justin Theroux Mim...
270	Karyn Kusama	Nicole Kidman	Sebastian Stan	Toby Kebbell	Thriller Crime Drama Action	destroyer	Nicole Kidman Sebastian Stan Toby Kebbell Kary...
271	David Slade	Fionn Whitehead	Will Poulter	Asim Chaudhry	Science Fiction Mystery Drama Thriller TV Movie	black mirror: bandersnatch	Fionn Whitehead Will Poulter Asim Chaudhry Dav...

272 rows × 7 columns

Figure 4.2.8: Final Dataframe for 2018 movies

Similarly, we do the extraction for movies in the years 2019, 2020,2021 and 2022 and then merge all the data frames and data from previous csv files to create a main file named “main_data.csv”.

	director_name	actor_1_name	actor_2_name	actor_3_name	genres	movie_title	comb
0	James Cameron	CCH Pounder	Joel David Moore	Wes Studi	Action Adventure Fantasy Sci-Fi	avatar	CCH Pounder Joel David Moore Wes Studi James C...
1	Gore Verbinski	Johnny Depp	Orlando Bloom	Jack Davenport	Action Adventure Fantasy	pirates of the caribbean: at world's end	Johnny Depp Orlando Bloom Jack Davenport Gore ...
2	Sam Mendes	Christoph Waltz	Rory Kinnear	Stephanie Sigman	Action Adventure Thriller	spectre	Christoph Waltz Rory Kinnear Stephanie Sigman ...
3	Christopher Nolan	Tom Hardy	Christian Bale	Joseph Gordon-Levitt	Action Thriller	the dark knight rises	Tom Hardy Christian Bale Joseph Gordon-Levitt ...
4	Doug Walker	Doug Walker	Rob Walker	unknown	Documentary	star wars: episode vii - the force awakens ...	Doug Walker Rob Walker unknown Doug Walker Doc...
...	...	...	...	...	...	...	...
6117	Pete Docter	Jamie Foxx	Tina Fey	Graham Norton	Animation Comedy Fantasy Family	soul	Jamie Foxx Tina Fey Graham Norton Pete Docter ...
6118	Robert Rodriguez	Priyanka Chopra Jonas	Pedro Pascal	YaYa Gosselin	Action Fantasy Family Comedy	we can be heroes	Priyanka Chopra Jonas Pedro Pascal YaYa Gossel...
6119	Regina King	Kingsley Ben-Adir	Eli Goree	Aldis Hodge	Drama	one night in miami...	Kingsley Ben-Adir Eli Goree Aldis Hodge Regina...
6120	Emerald Fennell	Carey Mulligan	Bo Burnham	Alison Brie	Thriller Crime Drama	promising young woman	Carey Mulligan Bo Burnham Alison Brie Emerald ...
6121	Kornél Mundruczó	Vanessa Kirby	Shia LaBeouf	Molly Parker	Drama	pieces of a woman	Vanessa Kirby Shia LaBeouf Molly Parker Kornél...

6122 rows × 7 columns

Figure 4.2.9: “main_data.csv”

4.2.1 Types of Data Set

For the review analysis of movies, we used the Naive bayes with a dataset split of 80% as training and 20% as testing. This gave the maximum accuracy of 98.7%.

4.2.2 Number of Attributes, fields, description of the data set

Our “main_data.csv” has 6122 data fields and has 7 features, which includes ‘director_name’, ‘actor_1_name’, ‘actor_2_name’, ‘actor_3_name’, ‘genres’, ‘movie_title’ and ‘comb’ (combination of features). This dataset has a list of almost all hollywood movies till the year 2022.

4.3 Design of Problem Statement

This movie recommendation system gives users a user-friendly interface as a website and it takes a movie name as input from the user and then shows the movie details, cast details, reviews of the movie and then the recommendation of similar types of movies.

4.4 Algorithm / Pseudo code of the Project Problem

- Takes movie name as input
- web scraping to get movie details from TMDB api and user reviews from IMDB site
- Then for recommendation, firstly we create a similarity matrix using the 'cosine_similarity' function imported from sklearn on the "main_data.csv" using the pairwise method.
- Then we find the movie title in the dataset.
- If the movie is not present in the dataset then we show an error message.
- Else we take the data of similarity matrix as a list of the movie.
- Then we sort the list and then create a list of movies which are closer to each other according to the cosine distance (we exclude 1st movie in list since it is the searched movie itself).

Our recommendation list is ready and is returned to the user.

Flow graph of the Minor Project Problem

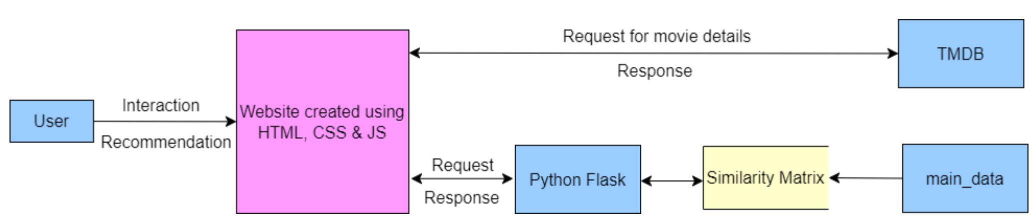

Figure 4.4.1: Flow chart of project

Screenshots of the various stages of the projects

Figure 4.4.2: Homepage

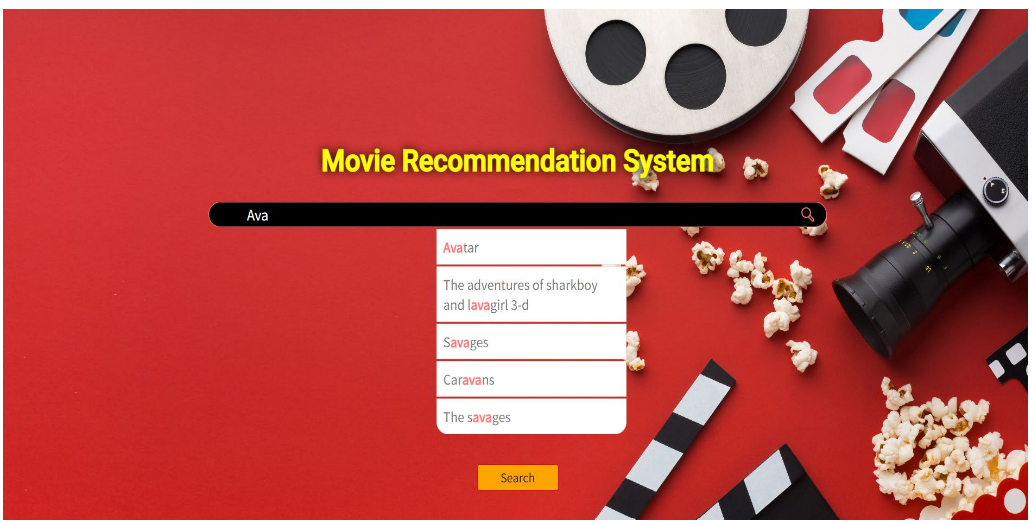

Figure 4.4.3: Auto Complete feature

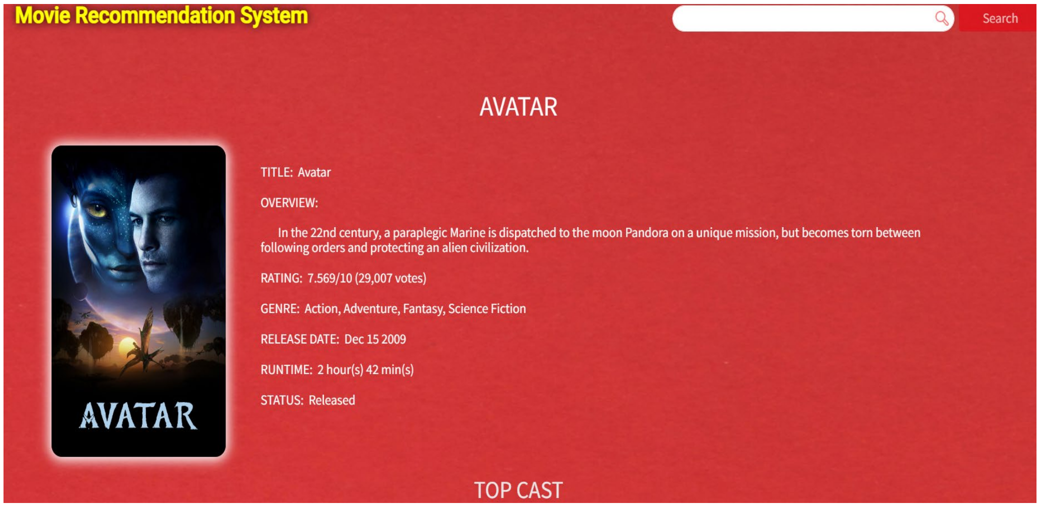

Figure 4.4.4: Movie Details

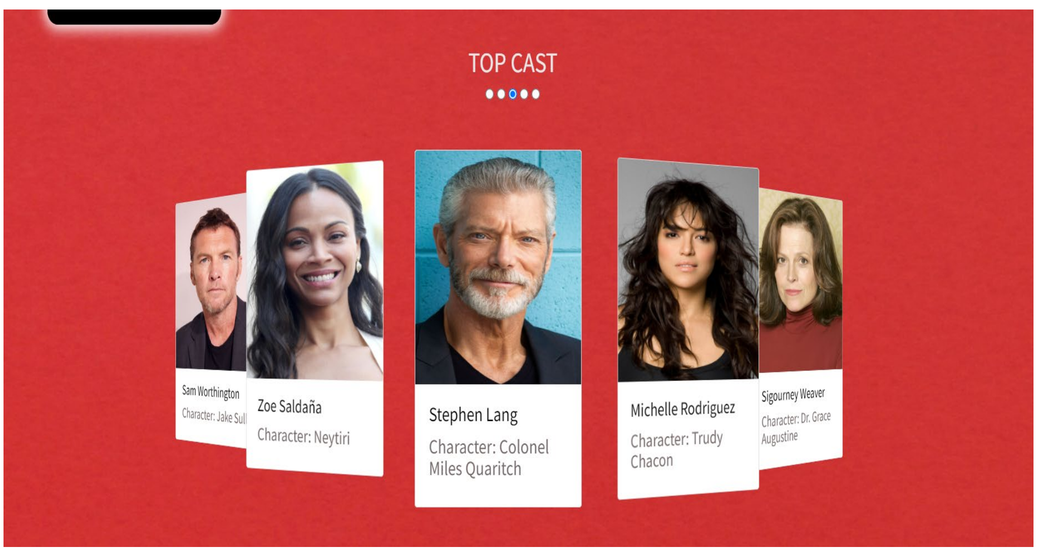

Figure 4.4.5: Cast information

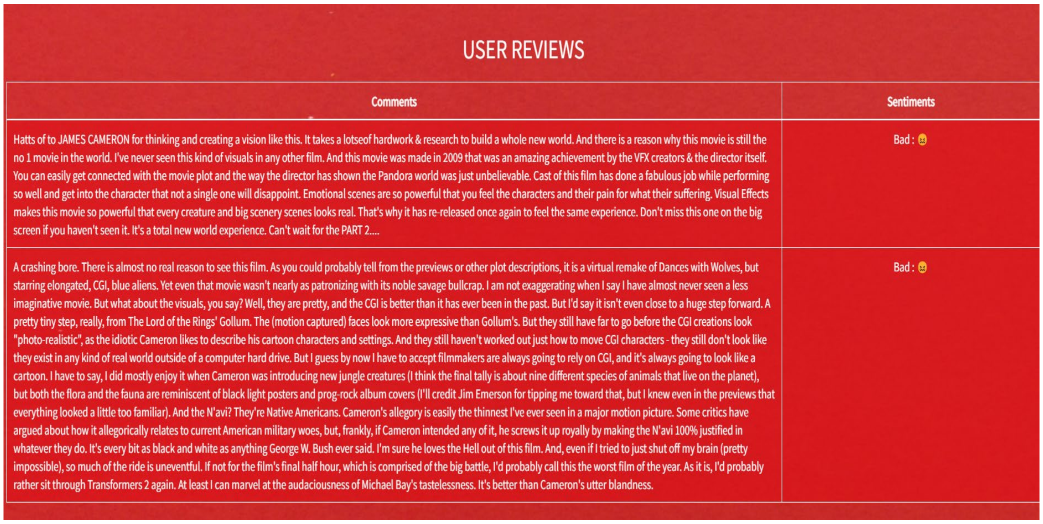

Figure 4.4.6: User reviews about the movie

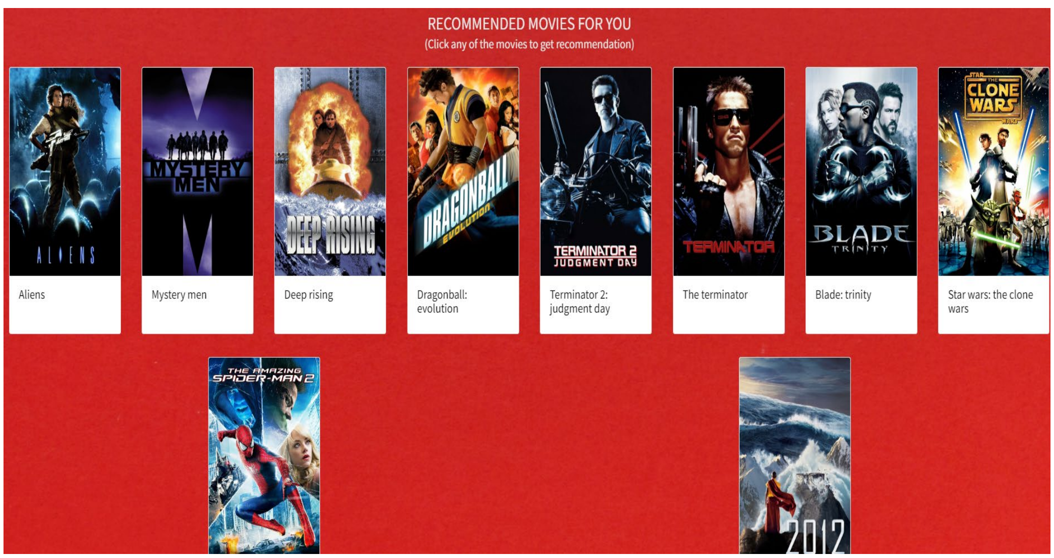

Figure 4.4.7: list of Recommended movies

CHAPTER 5: CONCLUSIONS

5.1 Conclusion

As per our aim and problem statement, our recommendation system is ready as a web page which is very user-friendly and easy to use. It takes a movie name as input and then it successfully matches it in our dataset to check for movie details and reviews. Movie details like cast and crew were taken from TMDB api and used IMDB website for getting reviews. Our recommendation engine successfully recommends movies on the basis of a similarity matrix generated from our dataset using cosine distance with features like genres and cast of the movies.

This system is viewed to users as a web page and the model works at the backend side. Movie name is passed in the backend and using the cosine similarity matrix, movies closest to the searched movie is recommended to the user.

5.2 Application of the Minor Project

The usage of recommendation systems is widespread and prevalent in practically all industries.These recommendation systems' principal objective is to help individuals save time. It therefore has a significant impact in a variety of sectors for this reason. Basically, real-world programmes like social networking, e-commerce, entertainment, and so forth utilise recommendation algorithms.

These recommendation systems are utilised in the entertainment industry whether watching films or listening to music. In essence, movie recommendation systems save users time by suggesting films to watch based on their interests rather than making them go through the time-consuming process of choosing a film to watch from a lengthy list or collection of films. For example if the person decides to indulge in a movie or series from an OTT platform then he/she ends up by watching more than what he/she has in their mind. So have you ever wondered how they could do such a thing? The reason is that almost all the OTT platforms rely on their movie recommendation systems. These OTT platforms identify the choice and priority of a user and show the best recommendation which has a higher chance to get selected by them.

High performance computing and machine learning advancements in recent years have sped up the growth of their respective industries.

5.3 Future Scope

Following aspects will we consider for the future work:

- Introduction of proper and more precise features of the film:

  Rating is employed by content-based filtering instead of object features, but in the future, object attributes like the movie's subtitles and colour should be extracted so that we can acquire more accurate data about the film.

- Introduce the list of movies which are disliked by the users:

  The information provided by users is usually helpful for recommendation systems, thus in the future we would gather additional information from users and then add a list of the things that consumers didn't like. Our objective at this point is to include this list into our recommender system and then produce the scores that will be added to our earlier findings. We can increase the accuracy of our recommender system in this way.

- Introduce the age of the user:

  The suggested method took the movie genre into account, but in the future we might also take the user's age into account. This is due to the fact that movie preferences vary depending on an individual's age; for instance, kids prefer animated films to other types of films. The proposed approach's memory needs will require more work in the future. The suggested method was applied here using simply the movie datasets, but it may also be applied using the Netflix datasets and the film affinity datasets.

REFERENCES

[1] Ahuja, R., Solanki, A., & Nayyar, A. (2019, January). Movie recommender system using k-means clustering and k-nearest neighbor. In *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)* (pp. 263-268). IEEE.

[2] Airen, S., & Agrawal, J. (2022). Movie recommender system using k-nearest neighbors variants. *National Academy Science Letters*, *45*(1), 75-82.

[3] Chauhan, A., Nagar, D., & Chaudhary, P. (2021, February). Movie Recommender system using Sentiment Analysis. In *2021 International Conference on Innovative Practices in Technology and Management (ICIPTM)* (pp. 190-193). IEEE.

[4] Cintia Ganesha Putri, D., Leu, J. S., & Seda, P. (2020). Design of an unsupervised machine learning-based movie recommender system. *Symmetry*, *12*(2), 185.

[5] Furtado, F., & Singh, A. (2020). Movie recommendation system using machine learning. *International journal of research in industrial engineering*, *9*(1), 84-98.

[6] Goyani, M., & Chaurasiya, N. (2020). A Review of Movie Recommendation System.

[7] Khatter, H., Goel, N., Gupta, N., & Gulati, M. (2021, September). Movie Recommendation System using Cosine Similarity with Sentiment Analysis. In *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)* (pp. 597-603). IEEE.

[8] Mehta, R., & Gupta, S. (2021). Movie Recommendation Systems using Sentiment Analysis and Cosine Similarity. *International Journal for Modern Trends in Science and Technology*, *7*(01), 16-22.

[9] Reddy, S. R. S., Nalluri, S., Kunisetti, S., Ashok, S., & Venkatesh, B. (2019). Content-based movie recommendation system using genre correlation. In *Smart Intelligent Computing and Applications* (pp. 391-397). Springer, Singapore.

[10] Singh, R. H., Maurya, S., Tripathi, T., Narula, T., & Srivastav, G. (2020). Movie recommendation system using cosine similarity and KNN. *International Journal of Engineering and Advanced Technology*, *9*(5), 556-559.

[11] Smitha, N., Anusha, D., Chaithanya, C., Sindhu, J., Tanuja, R., & Kumar, H. H. (2021, February). A Review on Movie Recommendation System Using Machine Learning. In *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)* (pp. 769-773). IEEE.

[12] Kaur, P., Singh, N., & Singh, K. (2021). Hybrid recommendation system using collaborative filtering and deep learning for movie recommendation. Journal of Ambient Intelligence and Humanized Computing, 1-10.

[13] Singh, R. K., & Tripathi, S. K. (2021). Movie recommendation system using enhanced KNN algorithm. IEEE Access, 9, 38078-38090.

[14] Ahmad, W., Javed, M. Y., Imran, M., & Hussain, I. (2021). A movie recommendation system based on semantic feature extraction and clustering. Journal of Ambient Intelligence and Humanized Computing, 1-15.

[15] Lee, J., Lee, Y., Lee, J., & Kwon, O. (2021). Movie recommendation system based on convolutional neural network with feature combination. IEEE Access, 9, 25983-25993.

[16] Singh, S., & Singh, S. (2021). An enhanced movie recommendation system using fuzzy clustering and cosine similarity. International Journal of Machine Learning and Cybernetics, 12(5), 1101-1113.

[17] Li, L., Li, J., & Zhang, Y. (2017). A hybrid recommendation algorithm based on deep learning. Neurocomputing, 275, 1214-1221.

[18] Yu, F., & Yang, J. (2019). A novel deep learning model for movie recommendation. IEEE Access, 7, 45003-45012.

[19] Zhang, X., Zhao, J., & Leung, H. F. (2020). Collaborative movie recommendation based on deep learning with hybrid autoencoder. IEEE Transactions on Neural Networks and Learning Systems, 32(8), 3513-3524.
[20] Zhang, X., & Leung, H. F. (2020). A hybrid deep learning model for movie recommendation. IEEE Transactions on Industrial Informatics, 16(7), 4432-4442.
[21] Liu, X., Wang, W., Wu, W., & Xu, H. (2021). An improved movie recommendation algorithm based on deep learning. IEEE Access, 9, 79238-79