**TRIBHUVAN UNIVERSITY INSTITUTE OF SCIENCE AND TECHNOLOGY**

**BIRENDRA MEMORIAL COLLEGE**

**A PROJECT REPORT**

**ON**

**“CONTENT BASED MOVIE RECOMMENDATION SYSTEM”**

**In the partial fulfillment of the requirement for the Bachelor’s Degree in Computer Science and Information Technology**

**Under the supervision of**

**Pravin Sangroula**

**Submitted By:**

**Bishal Giri [T.U Exam Roll no.: 24600/076]**

**Dhirendra B. Basnet [T.U Exam Roll no.: 24602/076]**

**Rojesh Chudal [T.U Exam Roll no.: 24607/076]**

**Submitted To:**

**Department of Computer Science and Information Technology**

**Birendra Memorial College**

**Dharan-1, Sunsari,**

**Nepal**

# LETTER OF APPROVAL

This is to certify that this project prepared by **BISHAL GIRI DHIRENDRA BAHADUR BASNET** and **ROJESH CHUDAL** entitled “**CONTENT BASED MOVIE RECOMMENDATION SYSTEM”** in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science and Information Technology has been well studied. In our opinion it is satisfactory in scope and quality as a project for the required degree.

|  |  |
| --- | --- |
| ……….…………………………  Pravin Sangroula  Supervisor  Birendra Memorial College | ………………………………  Mukesh Shah  HOD  Birendra Memorial College |
| ………..……..…………………… | ………………….………………. |
| (Upegdsfgsfdh) | (Kfggf) |
| External Examiner | Internal Examiner |
| IOST, Tribhuvan University | Birendra Memorial College |

# SUPERVISOR’S RECOMMENDATION

I hereby recommend that this project prepared under my supervision entitled

**“A CONTENT BASED MOVIE RECOMMENDATION SYSTEM”**

in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology be processed for the evaluation.

…………………………………………

Supervisor

Pravin Sangroula

# DECLARATION

We hereby declare that the project work entitled Content Based Movie Recommendation submitted to the Faculty of Science and Technology, Tribhuvan University, Kathmandu is an original piece of work under the supervision of (Mr. Pravin Sangroula) faculty members, Birendra Memorial college of IT, Dharan and is submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Science in Computer Science and Information Technology (B.Sc. CSIT). This project work report has not been submitted to any other university or institution for the award of any degree or diploma.

Submitted By:

Bishal Giri (24600/2076)

Dhirendra Bahadur Basnet (24602/2076)

Rojesh Chudal (24607/2076)

B.Sc. CSIT, 2076 Batch

Birendra Memorial College of Dharan

# ACKNOWLEDGMENT

The success and final outcome of this project required a lot of guidance and assistance from many people, and we feel extremely fortunate to have got all this all along the completion of our final year project. Whatever we have done is only due to such guidance and assistance and we would not forget to thank them. We could not have completed this project without help from our college “**BIRENDRA MEMORIAL COLLEGE**” which gave the supportive hands before us academically, also regarding other activities related to Information Communication Technology as well as extra curriculum activities that we got to participate in. It gave us family like environment.

We take this opportunity to express our profound gratitude and deep regards to our supervisor (**Mr.Pravin Sangroula)** for his exemplary guidance, monitoring, and constant encouragement throughout the course of this thesis.

We would also like to appreciate the help of various people for providing us essential data required for our project as well as their time in guiding us so as to how our project will be more effective in users prospective. We are also grateful to our teachers for their constant support and guidance.

Finally, we would like to express our sincere thanks to all the friends and others who helped us directly or indirectly during this project work. This project has been a wonderful experience where we have learnt and experienced many beneficial things.

With respect,

Bishal Giri (24600/2076)

Dhirendra Bahadur Basnet (24602/2076)

Rojesh Chudal (24607/2076)

# ABSTRACT

Recommendation based systems can be used for recommending different web page, books, restaurants, tv shows, movies etc. The aim of movie recommendation system is to recommend movies to different users based on their interests. This helps the user to save time browsing the internet looking for movies from the thousand already existing ones. Content-based recommendation system describes the items that may be recommended to the user. Based on a data set, it predicts what movies a user will like considering the attributes present in the previously liked movies. Recommendation systems can recommend movies based on one or a combination of two or more attributes. While designing a movie recommendation system various factors are considered such as the genre of the movie, the director or the actors present in it. In this paper, the recommendation system has been built on overview,cast, keywords, crew, and genres. A single column is created which will be the sum of all the 5 attributes, and it acts as a dominant factor for this movie recommender system.

Keywords*: Content based recommendation , PyCharm, Python, Machine learning, Web application.*

**TABLE OF CONTENTS**

[LETTER OF APPROVAL i](#_Toc161165945)

[SUPERVISOR’S RECOMMENDATION ii](#_Toc161165946)

[DECLARATION iii](#_Toc161165947)

[ACKNOWLEDGMENT iv](#_Toc161165948)

[ABSTRACT v](#_Toc161165949)

[LIST OF ABBREVIATIONS x](#_Toc161165950)

[LIST OF FIGURES xi](#_Toc161165951)

[LIST OF TABLES xii](#_Toc161165952)

[CHAPTER 1 1](#_Toc161165953)

[INTRODUCTION 1](#_Toc161165954)

[1.1 INTRODUCTION: 1](#_Toc161165955)

[1.2 PROBLEM STATEMENT: 1](#_Toc161165956)

[1.3 OBJECTIVES 2](#_Toc161165957)

[1.4 SCOPE AND LIMITATIONS 2](#_Toc161165958)

[1.4.1 SCOPE 2](#_Toc161165959)

[1.4.2 LIMITATIONS 3](#_Toc161165960)

[1.5 DEVELOPMENT METHODOLOGY 3](#_Toc161165961)

[1.6 REPORT ORGANIZATION 4](#_Toc161165962)

[CHAPTER 2 6](#_Toc161165963)

[BACKGROUND STUDY AND LITERATURE REVIEW 6](#_Toc161165964)

[2.1 BACKGROUND STUDY: 6](#_Toc161165965)

[2.2 LITERATURE REVIEW: 6](#_Toc161165966)

[CHAPTER 3 8](#_Toc161165967)

[SYSTEM ANALYSIS 8](#_Toc161165968)

[3.1 SYSTEM ANALYSIS 8](#_Toc161165969)

[3.1.1 REQUIREMENT ANALYSIS 8](#_Toc161165970)

[3.1.1 FEASIBILITY ANALYSIS 10](#_Toc161165971)

[3.1.2 ANALYSIS 11](#_Toc161165972)

[a. CLASS DIAGRAM 11](#_Toc161165973)

[b. OBJECT DIAGRAM 13](#_Toc161165974)

[c. STATE DIAGRAM 15](#_Toc161165975)

[d. SEQUENCE DIAGRAM 17](#_Toc161165977)

[e. ACTIVITY DIAGRAM 18](#_Toc161165978)

[4.1 DESIGN 19](#_Toc161165979)

[4.1.1 REFINEMENT OF CLASS DIAGRAM 21](#_Toc161165980)

[4.1.2 REFINEMENT OF STATE DIAGRAM 23](#_Toc161165981)

[4.1.3 REFINEMENT OF SEQUENCE DIAGRAM 24](#_Toc161165982)

[4.1.4 REFINEMENT OF ACTIVITY DIAGRAM 25](#_Toc161165983)

[4.1.5 COMPONENT DIAGRAM 26](#_Toc161165985)

[4.1.6 DEPLOYMENT DIAGRAMS 27](#_Toc161165986)

[4.2 ALGORITHM DESCRIPTION 28](#_Toc161165987)

[CHAPTER 5 31](#_Toc161165988)

[IMPLEMENTATION AND TESTING 31](#_Toc161165989)

[5.1 IMPLEMENTATION 31](#_Toc161165990)

[5.1.1 TOOLS USED 31](#_Toc161165991)

[5.2 TESTING 33](#_Toc161165992)

[TESTING METHODOLOGIES 33](#_Toc161165993)

[5.2.1.TEST CASES FOR UNIT TESTING 33](#_Toc161165994)

[5.2.2. TEST CASE SYSTEM TESTING 36](#_Toc161165995)

[TEST CASE NUMBER 1 36](#_Toc161165996)

[TEST CASE NUMBER 2 37](#_Toc161165998)

[CHAPTER 6 39](#_Toc161166000)

[CONCLUSION AND FUTURE RECOMMENDATION 39](#_Toc161166001)

[6.1 CONCLUSION 39](#_Toc161166002)

[6.2 FUTURE RECOMMENDATION 40](#_Toc161166003)

[REFERENCES 41](#_Toc161166004)

# LIST OF ABBREVIATIONS

|  |  |
| --- | --- |
| APP | Application |
| CBF | Content-Based Filtering |
| CSS | Cascading Style Sheets |
| HTML | Hyper Text Markup Language |
| IDE | Integrated Development Environment |
| IMBD | Internet Movie Database |
| JS | JavaScript |
| OS | Operating System |
| PY | Python |
| PIP | Preferred Installer Program |
| TMDB | The Movie Database |

# LIST OF FIGURES

[Figure 1.1 : Incremental Model 4](#_Toc161166871)

[Figure 3.1 **:** Use Case Diagram 9](#_Toc161166872)

[Figure 3.2 : Class Diagram 12](#_Toc161166873)

[Figure 3.3 : Object Diagram 13](#_Toc161166874)

[Figure 3.4 : State Diagram 15](#_Toc161166875)

[Figure 3.5 : Sequence Diagram 17](#_Toc161166876)

[Figure 3.6 : Activity Diagram 18](#_Toc161166877)

[Figure 4.1 : Attribute used from dataset 19](#_Toc161166878)

[Figure 4.2: Resulting attribute after combining 5 attributes 20](#_Toc161166879)

[Figure 4.3: Movies in the form of vectors 20](#_Toc161166880)

[Figure 4.4 : Recommendation by model based on cosine similarity 21](#_Toc161166881)

[Figure 4.5: Class Diagram for a Proposed System 22](#_Toc161166882)

[Figure 4.6 : State Diagram 23](#_Toc161166883)

[Figure 4.7 : Refinement of Sequence Diagram 24](#_Toc161166884)

[Figure 4.8 : Activity Diagram 25](#_Toc161166885)

[Figure 4.9 : Component Diagram 26](#_Toc161166886)

[Figure 4.10 : Deployment Diagram 27](#_Toc161166887)

[Figure 4.11 : Cosine Similarity Formula 28](#_Toc161166888)

[Figure 4.12 : Movie vectors representation 29](#_Toc161166889)

# LIST OF TABLES

[Table 5.1 : Test cases for interface 34](#_Toc161166914)

[Table 5.2 : Test Cases for Model Accuracy 35](#_Toc161166915)

[Table 5.3 Test Cases for Performance 35](#_Toc161166916)

[Table 5.4 : System testing 37](#_Toc161166917)

# CHAPTER 1

# INTRODUCTION

## INTRODUCTION:

Recommendation System is an information tool which helps users to find out the items which they want from the large number of items available. The recommendation system excavates the resources that users may be interested in or need according to the interest characteristics of different users from the mass information and makes recommendations [[1](#RNB16)]. Certainly! In today's world, recommendation systems are crucial in platforms like YouTube, Netflix, Amazon Prime, Instagram, and Facebook. These systems suggest items to users based on their preferences or ratings given to items. By predicting user preferences, they recommend items that align with a user's interests, aiding in suggesting things they might like when browsing these websites. For instance, if you search for a mobile phone on Amazon or Flipkart, the next time you visit the app, you might see related mobile phones based on your previous search. This recommendation process also occurs in streaming platforms like Amazon Prime and Netflix, where movies are suggested based on a user's history and ratings. Everything a user does on these platforms is observed by a system, enabling personalized recommendations. These recommendations often utilize filtering techniques: content-based filtering, collaborative filtering, and hybrid filtering. In our project we have used content-based filtering with the TMDB movie dataset and developed using Flask and Python libraries like pickle, sklearn, NumPy, pandas and requests, aims to build a model that simplifies movie recommendations based on user behavior.

## PROBLEM STATEMENT:

The goal of the project is to recommend top 5 movies to the user according to the what the he or she wants to watch which also saves up a lot of time for the user.

Also, Providing related content out of the relevant and irrelevant collection of items to users of online service providers.

## OBJECTIVES

A content-based recommendation system’s goal is to provide recommendations for products or content to consumers based on the features and attributes of the products as well as the user’s previous choices or actions [[2](#Bil98)]. Put differently, the algorithm suggests products based on their content similarity to those the user has previously enjoyed or engaged with. The objective of the project can be stated as:

* To provide each user customized suggestions based on their unique preferences, interests, and behavior.
* To enhance the user experience by facilitating the discovery of fresh material that corresponds with the interests of the user.
* To provide a variety of choices in order to steer clear of repetitious suggestions and expose customers to a greater selection of products.

Since the suggestions are based on item features rather than user behavior, they may be helpful for new users who haven’t interacted with the platform before.

## SCOPE AND LIMITATIONS

### SCOPE

The different objectives of this project are as follows:

Movie Features: Content Based Movie Recommendation System should be able to analyze attributes like genre, director, actors, release year, and more to suggest similar movies based on these characteristics.

User Profiling: Content Based Movie Recommendation System should be able to create personalized user profiles by understanding their movie preferences, viewing, history, ratings, and explicit feedback.

Algorithms Approach: Content Based Movie Recommendation System should be able to use machine learning or AI algorithms to compare movie attributes and user preferences, often employing techniques like cosine similarity.

Personalization: Content Based Movie Recommendation System should be able to offer tailored recommendations to users by understanding individual tastes and preferences.

Challenges: Content Based Movie Recommendation System should be able to address issues like the cold-start(for new users or movies with few ratings), ensuring diversity in recommendations and maintaining accuracy despite sparse or noisy data.

### LIMITATIONS

The different limitations of this project are as follow:

The system often recommend movies similar to ones already liked, which may limit the discovery of diverse or unexpected choices.

The system relies heavily on accurate and detailed metadata such as genre, and actors which might not always capture the nuanced aspects that appeal to users.

The system may face difficulty in recommending movies for new users or newly released films due to a lack of sufficient data or ratings.

The system doesn’t consider social interactions or recommendations from friends, which can significantly impact movie choices.

The system often doesn’t adapt quickly to sudden shifts in user preferences or trends since it primarily relies on historical data.

## DEVELOPMENT METHODOLOGY

The methodology used for this project is called incremental delivery, and it divides requirements into several independent software development cycle modules. The phases of incremental development are analysis, design, implementation, testing, and maintenance. The phases of requirements, design, coding, and testing are completed in each iteration. And until all planned functionality is implemented, every new release of the system builds upon the functionality of the one before it. As soon as the first increment is delivered, the system is put into production. The initial iteration typically consists of a core product that fulfills fundamental needs; additional features are added in subsequent iterations. The next increment’s plan is developed once the client has evaluated the core product.

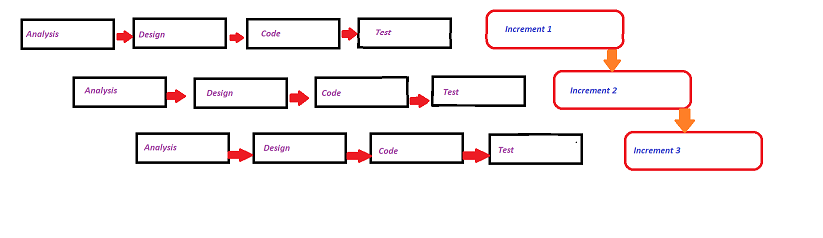


Figure 1.1 : Incremental Model

## REPORT ORGANIZATION

The report has been prepared following the guidelines provided by Tribhuvan University. The report is separated into different chapters. Each chapter consists of various sub chapters with its content. The preliminary section of the report consists of Title Page, Acknowledgement, Abstract, Table of Contents, List of Abbreviations, List of Figures, and List of Tables.The main report is divided into 6(six) chapters which includes:

Chapter 1: Introduction

It includes an overall summary of the project and the system. It contains

the system’s development methodology, scope and limitations, and problem statement.

Chapter 2 : Background Study and Literature Review

It includes background study, literature review section with a detailed discussion of the system-related research projects.

Chapter 3 : System Analysis

It includes requirement analysis, function requirements, non-functional requirements, and feasibility analysis. Additionally, an object-oriented or structured UML diagram is used.

Chapter 4 : System Design

It includes algorithmic details as well as system design. Various diagrams are displayed, such as class, object, and activity diagrams.

Chapter 5 : Implementation and Testing

It includes details on the testing and implementation procedures. We also talk about the tools that are utilized in this chapter.

Chapter 6 : Conclusion and Future Recommendations

It includes an overview of the system and the project in its entirety. It also contains the essential suggestions and the project’s future scope.

# CHAPTER 2

# BACKGROUND STUDY AND LITERATURE REVIEW

## BACKGROUND STUDY:

System for recommendations are becoming more and more crucial in the incredibly hectic world of today. Due to the numerous tasks that people have to complete in the limited 24-hour period, people are never on schedule. Recommendation structures are so essential since they enable people to make wise decisions without depleting their cognitive capacity. A recommendation system’s primary function is to find content that would excite a particular user. Additionally, it has several features that allow for the creation of personalized lists of interesting and helpful stuff for each user of individual. Artificial intelligence-based algorithms, known as recommendation structures, go through all of the options and produce a personalized list of items that may be interesting and pertinent to a particular person.

## LITERATURE REVIEW:

Recommender systems are ones that can make suggestions, make playlists, locate matches, and do a lot more. The operation of recommender systems relies heavily on characteristic information and user-item interactions. While information regarding user-item interactions includes ratings, the amount of purchases made, user likes and many other things, characteristic information is information about the person and the objects. This allows the recommendation system to be created using a collaborative filtering, content-based filtering, or hybrid filtering method.

1. **Collaborative Filtering**:

The concept of collaborative filtering was first introduced in 1991 by Goldberg et al [[3](#Gol01)].This algorithm looks for users with similar tastes, then leverages their comments to recommend comparable content to other users who also have those interests. It generates recommendations by using information from rated profiles for different people or objects. Numerous programs have integrated it, such as YouTube, Netflix and Spotify. It’s a standard tactic and a part of the hybrid system. collaborative filtering focuses on identifying users with similar preferences and generating recommendations

based on the preferences of these similar users [[4](#RHS20)].

1. **Content-Based Filtering:**

Cognitive filtering is another name for Content-Based filtering . In content-based recommender systems, all the data items are collected into different item profiles based on their description or features. For example, in the case of a book, the features will be author, publisher, etc. In the case of a movie, the features will be the movie director, actor, etc. [[5](#Roy22)] Techniques for content-based filtering are informed by user attributes. This method is used when details about an item, like its name, location, or description, are known but not about the user. Similar to collaborative techniques, it makes assumptions based only on user information and completely ignores feedback from other users. It uses the data supplied by the user, either explicitly or implicitly. The more content-based filtering tools that users supply, including content-based recommenders, the more accurate the engine becomes.

1. **Hybrid Approach:**

A Hybrid approach is one that combines content-based filtering, collaborative filtering, or any other strategy. This filtering is an information filtering system that takes ratings of the movies as input from the users and then apply the collaborative and content based filtering and generate recommendation list [[6](#RHS201)]. It raises the accuracy and performance of recommender systems.

# CHAPTER 3

# SYSTEM ANALYSIS

## SYSTEM ANALYSIS

Analyzing a system or organization to determine its constituent parts, their interactions, and area for improvement is the process known as systems analysis. With a focus on the system as a whole and the connections between its components, it is a holistic approach. Identification of issues and inefficiencies in the current system, along with the suggestion of improvements, is the aim of systems analysis. It consists of the feasibility study, the analysis and the functional and non-functional needs. What the system should be able to perform, including the services it should offer and the limitations on how it can operate, is described by requirements analysis. The system’s usability and performance are characterized by its functional needs. All project related aspects are taken into account in a feasibility analysis.

### REQUIREMENT ANALYSIS

It is a component of project management that aids in guaranteeing relevance, clarity, and completeness. Setting expectations for a project is the aim. The needs could be related to a new technology purchase, process enhancement, or software development. Requirements analysis comes in several forms.

1. **FUNCTIONAL REQUIREMENTS**

* **User Input:** Allow user to input their best preference of movie
* **Similarity calculation:** Calculate the similarities between movies based on cosine similarity.
* **Recommendation generation:** Recommend the best movies taking care of similar attribute like genre, cast and crew, director.

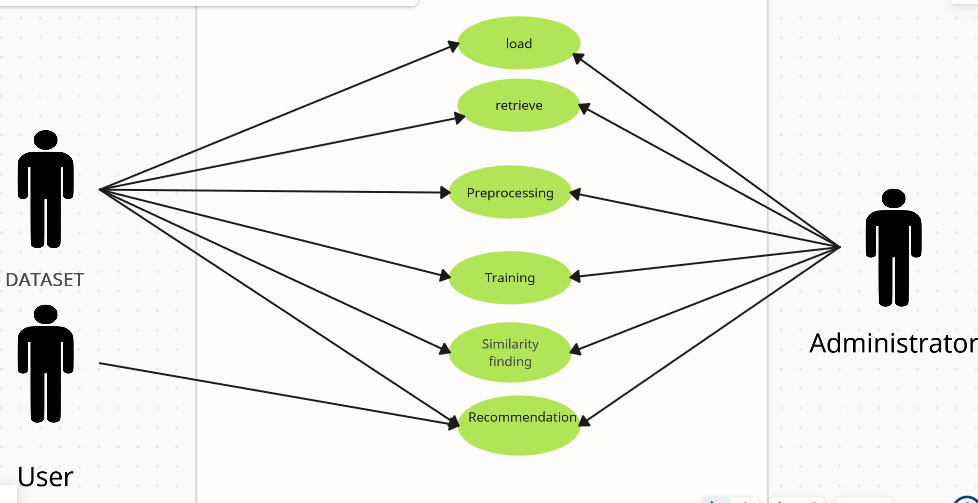
****

Figure 3.1 **:** Use Case Diagram

1. **NON-FUNCTIONAL REQUIREMENTS**

Non-functional requirements are performance, usability, accuracy and maintainability.

* **Performance:** This system recommends movies with in few milliseconds to seconds providing best performance.
* **Usability:** This system is provided with good and simple user interface components like buttons and select to search for the movies.
* **Accuracy:** This recommendation system provides high quality recommendation to the user based on the input provided by them.
* **Maintainability:** Code and documentation of this recommendation system is clean and well maintained which ensures that futures modification in a system will be most easier and faster.

### FEASIBILITY ANALYSIS

1. **TECHNICAL FEASIBILITY:**

Examines whether the new system can be developed and implemented using existing technology**.** [[7](#Con03)]In our project the system was mainly build using python programming language with some HTML, CSS and JavaScript. This system was developed under flask following proper standard. Flask is open sourced framework which allows our system for easy integration with other code libraries and extensions. Hence our system was known to be technically feasible.

**ii. OPERATIONAL FEASIBILITY:**

Operational feasibility is a measure that shows the capability of proposed system to solve problem and to take advantage of opportunities that arises on the course of performing day to day task. Since was proposed system is accurate, reliable, complete and affordable. Also due to the simplicity of flask, we can easily maintain and update our system under this framework. So this system is operationally feasible.

1. **ECONOMIC FEASIBILITY:**

Economic feasibility is something that has to be done before starting a project. IT analyse cost of developing the project and total revenue it will be generating after being developed. Cost for developing the project was minimal as flask framework was open sourced framework and Development team just make a use of their own personal mobile, computer and internet for development. Moreover, if we want to add some additional feature to our project then it will be free of cost and no additional equipment will be needed for that. Updated and extensive database can be maintained with increased workload and number of users.

1. **SCHEDULE FEASIBILITY:**

This part of feasibility analysis is most significant among all. Project must be completed in time regardless the complexity and size of the system to be developed. The proposed project along with its related activities was completed within following time constraints.

### ANALYSIS

### CLASS DIAGRAM

Class diagram illustrates the relationship and dependencies among class in Unified Modeling Language (UML). Class diagrams are also the foundation for a couple of related diagrams: component and deployment diagrams. [[8](#Jan17)]

The main purpose of class diagrams is to build a static view of an application. It is the only diagram that is widely used for construction, and it can be mapped with object-oriented languages. It is one of the most popular UML diagrams.

In below diagram , all the attributes related to the movie are stored in Movie class. MovieDataset manages the collection of the movies . It also provides method to add and get the movie .Model and Similarities are dependent upon MovieDataset and RecommendationSystem class uses a Model and MovieDataset to recommend movies.UserInterface class shows the interface to the user with poster , cast ,crew, release date and other movie details.

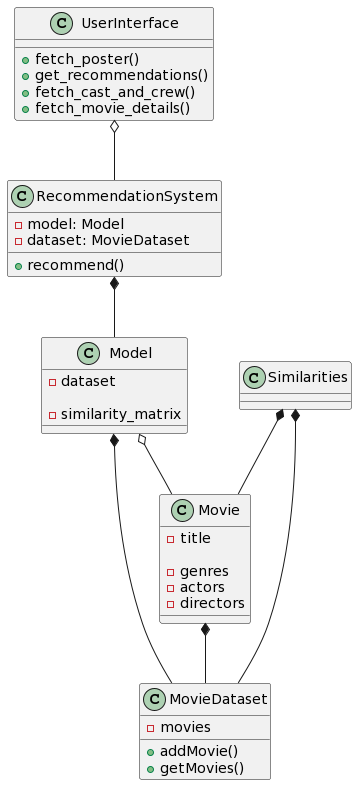


Figure 3.2 : Class Diagram

### OBJECT DIAGRAM

Object diagram are subset of class diagram which are instances of class diagram. The purpose of object diagram vastly reflects the system behavior.

Class diagram provides a static view of class and model whereas in other hand object diagram provides concrete view of state of system at a particular instance.

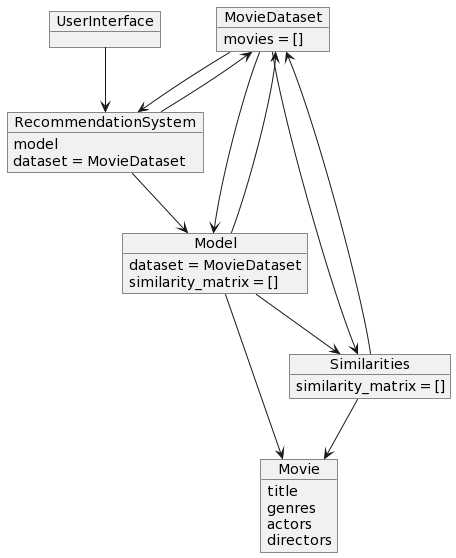


Figure 3.3 : Object Diagram

Above object diagram contains list of many classes like UserInterface , MovieDataset ,RecommendationSystem , Model , Movie , Similarities. These classes contains different methods and attributes .

The Movie and MovieDataset classes handle the representation and management of movie data. The Model class serves as a central component for the recommendation system, utilizing a MovieDataset object for data and potentially computing similarities between movies. The RecommendationSystem class orchestrates the process of generating recommendations, leveraging the Model and MovieDataset objects to provide personalized recommendations to users. The UserInterface class provides a user-friendly interface for users to interact with the recommendation system, facilitating tasks such as fetching movie details and receiving recommendations. The Similarities class assists in computing similarities between movies, contributing to the recommendation generation process.

### STATE DIAGRAM

### State diagram describes the behavior of system that describes the condition of system at finite instance of time. A state diagram consist of states, transitions, and activities. Overall, state diagram represents the dynamic view of system.

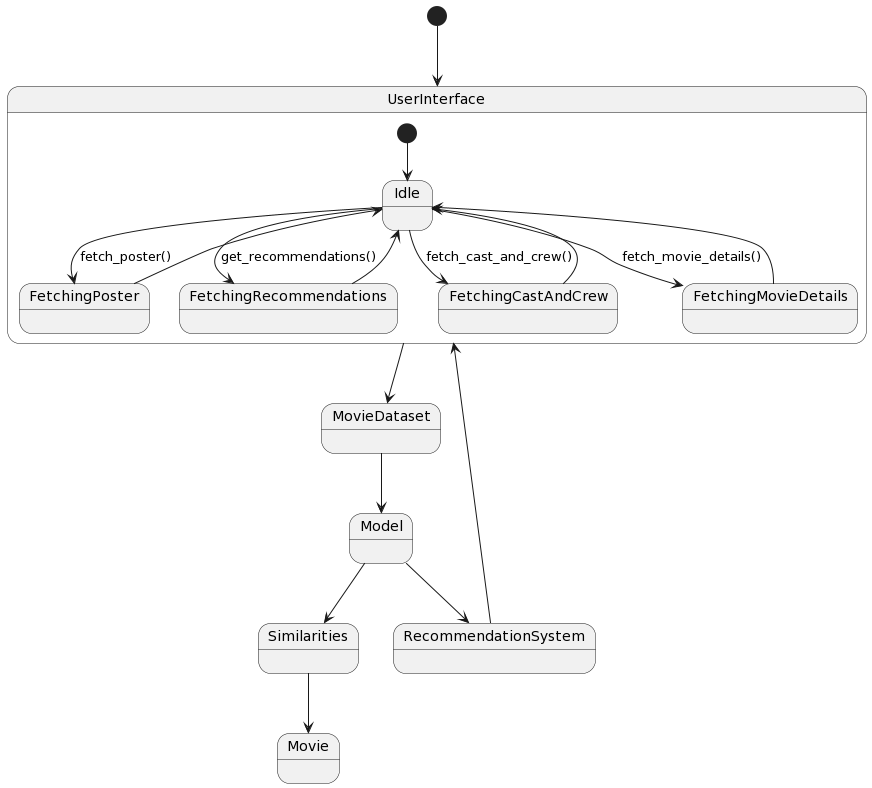


Figure 3.4 : State Diagram

Above state diagram consist of many states i.e. FetchingPoster, FetchingRecommendations, FetchingCastAndCrew,FetchingMovieDetails,MovieDataset,Model,Similarities,RecommendationSystem and Movie .There are several transition between these states .Like in initial our sytem is in idle state and it transition to FetchingPoster state with fetch\_poster() event .Also to FetchingRecommendations state with get\_recommendations() event and so on.

In abstract view, UserInterface will be transitioned to backend and then backend will be performing all the processing function and transitioned back to the UserInterface .Within UserInterface, there are many state and many transition are being carried out in between them to produce the desired recommendation.

### SEQUENCE DIAGRAM

Sequence diagram is used primarily to show the interactions between objects in the sequential order that those interactions occurs. An organization’s technical staff can find sequence diagrams useful in documenting how a future system should behave. [[9](#Bel04)] It represents the sequence of messages exchanged between objects or components. Sequence diagram typically consist of objects, lifeline, messages, activation box and some optional elements like look fragments, alternative fragments and other control flow constructs.

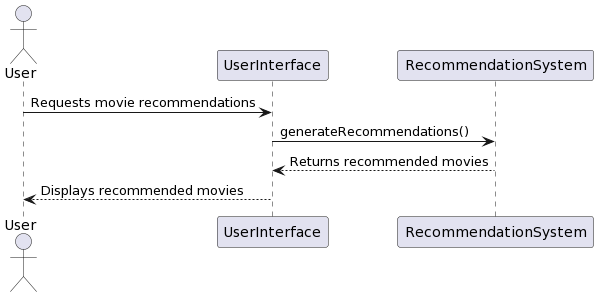


Figure 3.5 : Sequence Diagram

Above sequence diagram shows the illustration of sequence of message exchanged between User and proposed system over time. Here, objects are User , UserInterface and RecommendationSystem. Message exchanged between objects are mentioned with the arrow between the them.

User make a request for movie recommendation to the system through its UserInterface. After this, UserInterface makes call to the backend of our proposed system for generating recommendation and it returns the list of recommended movies.The user interface fetches the detail for recommended movies and displays the result to the users.

### ACTIVITY DIAGRAM

Activity diagram simply represents the flow from one activity to another activity. Activity is a particular operation of the system. It is also sometime called as object oriented flowchart. This model is widely used to model flow of work in a system showing the actions which are performed in sequence.

The main theme of activity diagram is to capture the dynamic behavior of the system. Activity diagram lacks message part.

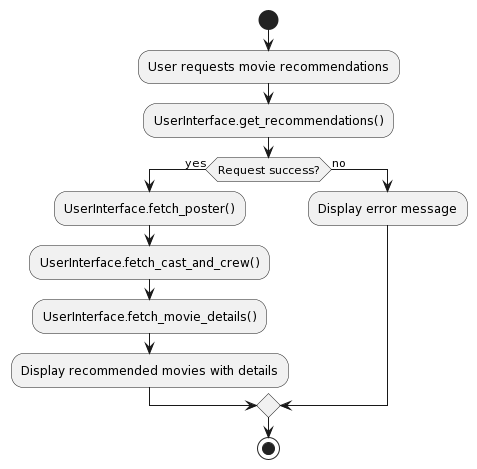


Figure 3.6 : Activity Diagram

In above activity diagram , user requests a movie recommendation by giving the input in user interface. If request is success the user interface fetches poster, cast ,crew and other related movie details and shows the movie with all the fetched details. If request is not success then system simply displays error message.

**CHAPTER 4**

**SYSTEM DESIGN**

## DESIGN

System design is a process of defining the architecture, components and interactions of a software system to fulfill specific functional and non functional requirements.

Alternatively, system design is considered blueprint that maps concepts and requirements into tangible software structure.

In this phase,we collected a secondary dataset from Kaggle dataset i.e. tmdb\_5000\_movies.csv and tmdb\_5000\_credits.csv .There were 4804 tuples in tmdb\_5000\_movies.csv dataset where as in there were 4814 tuples.In first dataset there were 22 attributes.This data is in raw form so we perform data processing to refine this data.

* **Data Processing**

In data processing, we remove most of the non useful attributes resulting 7 useful attribute in dataset i.e. movie\_id, title, overview, genres, keywords, cast and crew.

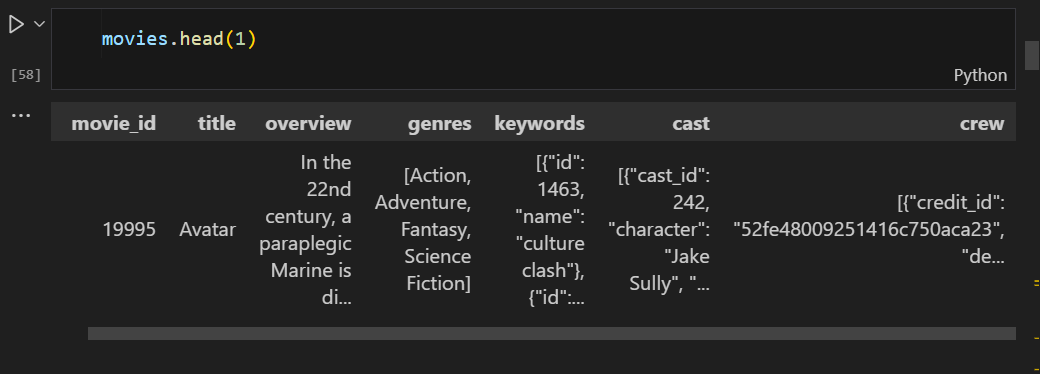
**

Figure 4.1 : Attribute used from dataset

The value of above attributes were converted to usable format and only value for director was fetched from crew attribute.Then 5 attributes (overview,genres,keywords,cast and crew) were combined to make single attribute “tags”. The resulting output is shown in below figure:



Figure 4.2: Resulting attribute after combining 5 attributes

After this we perform stemming where all the word of same meaning were transformed to simple word like run,running,runnable were converted to run.Also all the stop words were removed and all the word were transformed to lower case. Then we apply text vectorization to the resulting output.The output of data processing contains 4806 tuples and 3 columns.

* **Text Vectorization**

In text vectorization,each tuples in the above resulting dataset is converted in the form of vectors by applying CountVectorizer method and are stored in form of array.The resulting array is given as:

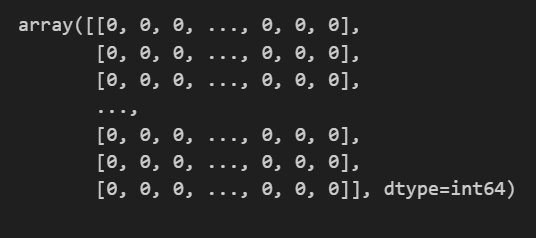


Figure 4.3: Movies in the form of vectors

Again,cosine similarities between each movie were calculated and recommendation were made on the basis of the cosine similarity which means that only those movie having less cosine similarity with the particular movie were recommended to the user.

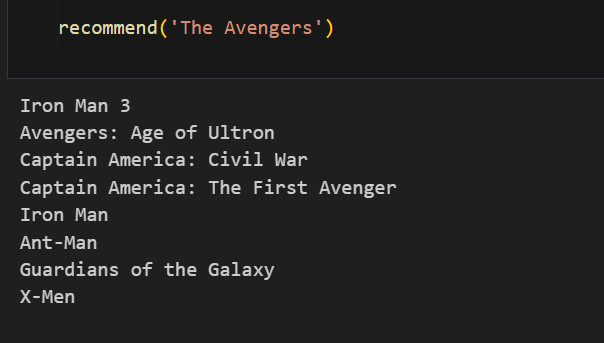


Figure 4.4 : Recommendation by model based on cosine similarity

### REFINEMENT OF CLASS DIAGRAM

Refined class diagram provides more detailed representation of the system component which makes us easier to dive to overall working of system. In these refined class diagram, methods names and parameters are more clarified for better understanding and also some additional methods and attributes are added to enhance in-depth understanding.

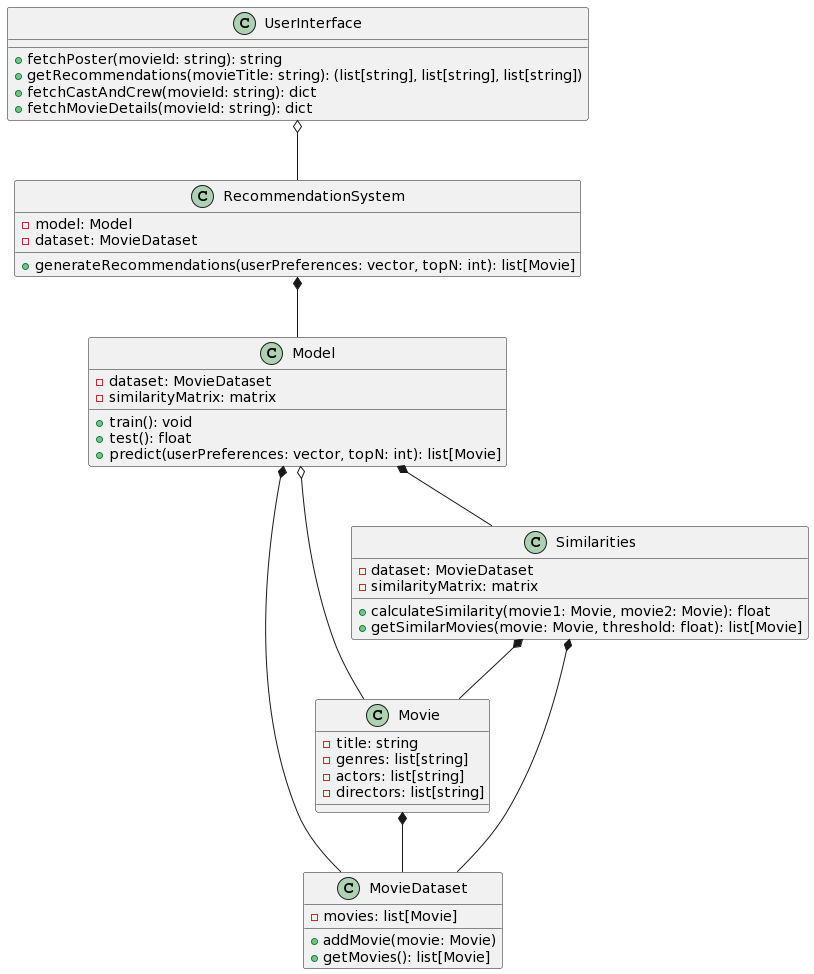
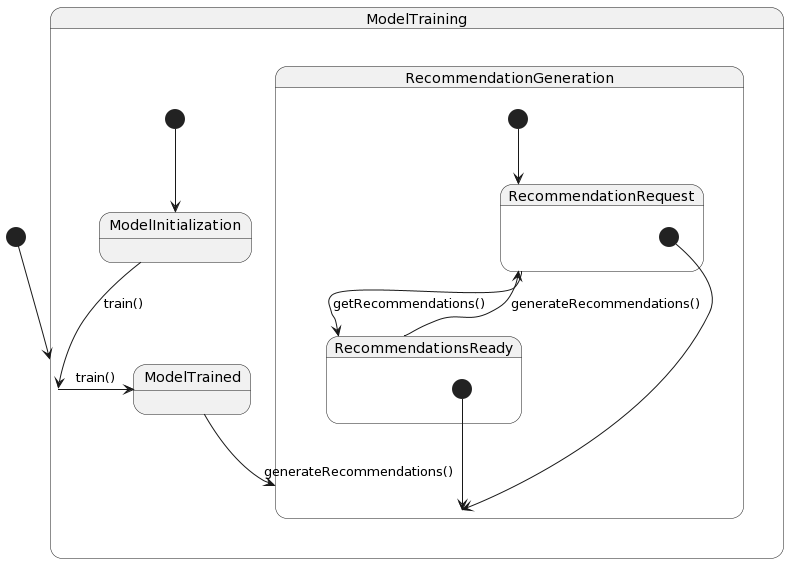


Figure 4.5: Class Diagram for a Proposed System

### REFINEMENT OF STATE DIAGRAM

Refined state diagram provides more concrete representation of description of condition of system at finite instance of time. In below refined state diagram, the system starts in the ModelTraining state and goes through ModelInitialization and in ModelTrained state out model is trained. After ths state our system can generate recommendationGeneration. Also recommendations can be requested using RecommendationsReady once recommendation are prepared.

Figure 4.6 : State Diagram

### REFINEMENT OF SEQUENCE DIAGRAM

A refined sequence diagram is a diagram that provides detail representation of sequence of messages exchanged between objects or components.

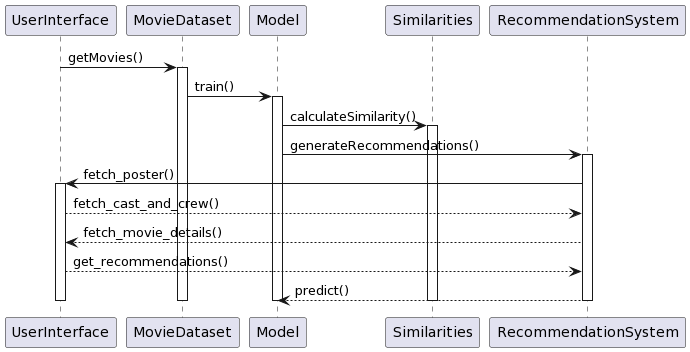


Figure 4.7 : Refinement of Sequence Diagram

### REFINEMENT OF ACTIVITY DIAGRAM

### The refined activity diagram provides more detailed representation of flow between activities. This system starts by receiving a request for recommendations. It checks if the model is trained. If yes then it generated recommendation and if no then it trains the model first and generated recommendations. After generating the recommendations, it checks if they successfully generated. If yes, it displays the recommendations and if not then it notifies the user on an error.

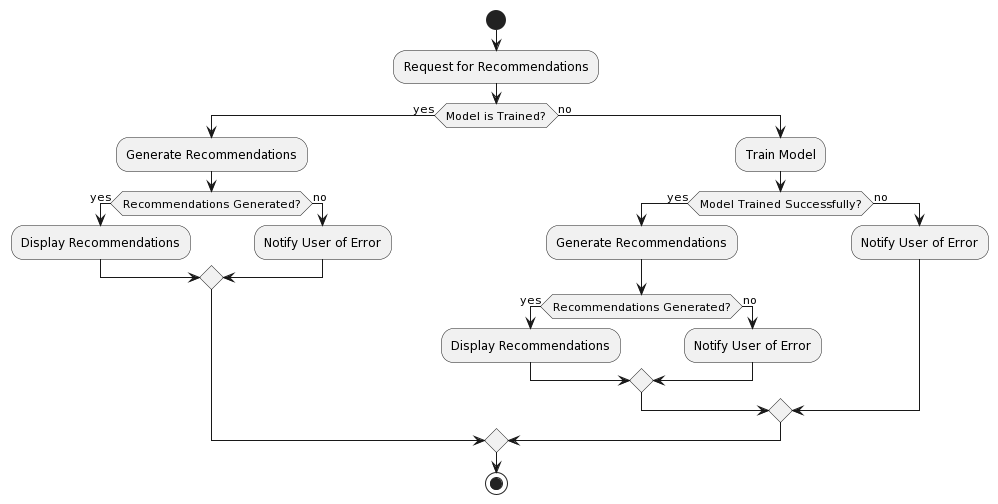


Figure 4.8 : Activity Diagram

### COMPONENT DIAGRAM

A component diagram is used to decompose a large system into smaller subset that visualizes the relationships as well as the organization between the components present in the system. Components present in the system. Components constitute of individual modules, libraries, executables, or other parts within a system.

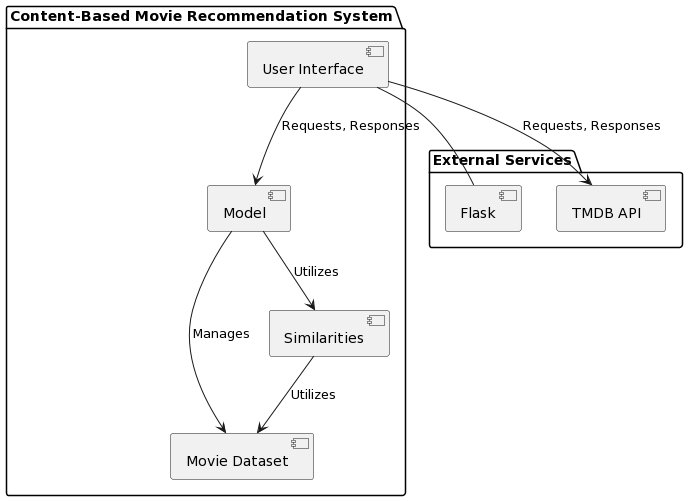
Incorporating all of these we can say that a component diagram depicts how components are wired together to form larger components or software systems.

Figure 4.9 : Component Diagram

### DEPLOYMENT DIAGRAMS

Deployment diagram in Unified Modeling Language (UML) helps in modeling of the physical aspect of the system. The diagram shows what hardware and software are necessary if you want to deliver the intended solution. It also shows how everything connects to form the system that you’ll use for delivery.

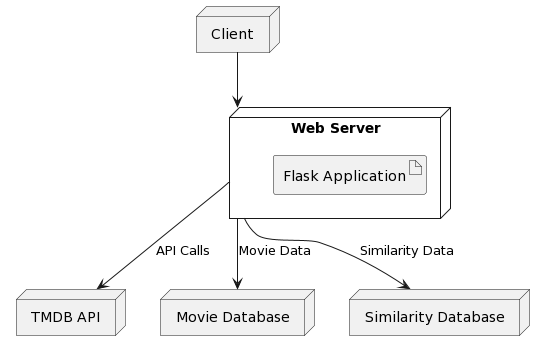


Figure 4.10 : Deployment Diagram

## ALGORITHM DESCRIPTION

**Count Vectorizer**

Machines cannot understand characters and words So when dealing with text data we need to represent it in numbers to be understood by the machine. Count vectorizer is a method to convert text to numerical data. Count vectorizer is a feature extraction technique that is commonly used in Natural Language Processing (NLP) and text minin. This technique is used to convert the text documents into a matric of token count. Count vectorizer makes it easy for text data to be used directly in machine learning and deep learning models such as text classification.

In this project, we derived the vector for each movie and find the cosine similarity between each movie. After this we recommend the movies to the user that the smallest cosine similarity to the movie that user is searching for.

**Cosine Similarity**

Cosine similarity is a measurement that quantifies the similarity between two or more vectors for movies. It’s the cosine of the angle between vectors, which are typically non-zero and within an inner product space. . This approach facilitates the efficient and accurate generation of recommendations in the research context [[10](#XSu09)]. Cosine similarity is a commonly used similarity measurement technique that can be found in libraries and tools such as Matlab, Scikit-Learn and TensorFlow, etc.

Let us assume two vectors A and B. If we have similarity measurement between A and B as 1 then it means that two vectors are aligned in the same orientation. Also if the similarity measurement between two vectors is 0 then it means that two vectors are aligned perpendicularly. If similarity measurement is -1 then it means that they are oriented in exactly opposite directions (back to back). The cosine similarity formula can be mathematically described as shown below.



Figure 4.11 : Cosine Similarity Formula

A.B = Dot product between the two movies vectors,

||A|| ||B|| = Product of the magnitudes of the two movie vectors

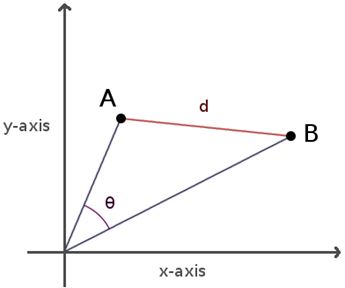


Figure 4.12 : Movie vectors representation

As shown in Fig, the area below the movie vectors A and B, represent the contents of the movies, and the angle, 0, between the movie contents. Thus, the lower the angle 0,

the more similar the movie contents are.

The cosine similarity values can range between 0 and 1, depending on the 0 value bounded between 0 and 90.

For the implementation of above algorithm following component are used:

**Programming Language**:

Python

Libraries

Pandas

Numpy

Requests

Pickle

**Content-Based Filtering algorithm**

Content-Based filtering methods are based on user characteristics; they are used when information about an item, such as its name, location, or description, is known and not about the user; they predict the items based on user information and, unlike collaborative techniques, completely ignores contributions from other users. The content-based filtering methods use the date that the user provides, either explicitly or implicitly; the more content-based filtering mechanisms actions the recommendations receive, such as content-based recommender, the more accurate the engine becomes. In simple word, we may receive movie suggestions based on descriptions of previous films. We trained the machine using a variety of datasets from various sources such as Kaggle, so that it could distinguish between movies and their specifics, such as genre, synopsis, and so on.

# CHAPTER 5

# IMPLEMENTATION AND TESTING

# 5.1 IMPLEMENTATION

# 5.1.1 TOOLS USED

**CASE TOOLS** : CASE tools are software applications specifically designed to assist in various stages of the software development lifecycle, such as requirements gathering, analysis, design, implementation, testing, and maintenance. There is no any explicit CASE tool in a system. But some tools are used as provided below:

1. **NLTK (Natural Language Toolkit)** : These tools are used text preprocessing tasks. These includes tokenization, removing stop words, stemming and converting all the word to lower cases.
2. **DE (Integrated Development Environment)**: These system was implemented using tools like PyCharm, Visual Studio Code and Anaconda in writing and managing the code effectively.
3. **Version Control System (VCS)**: We have used VCS such as Git along with platforms like GitHub which facilitate us collaborative development, enabling multiple member in team to work on the project simultaneously.
4. **PlantUML** : This a tool which was used for creating various types of diagrams, including class diagrams, sequence diagrams, activity diagrams, use case diagrams, and more.

**PROGRAMMING LANGUAGE :** Following programming language were used in implementation of movie recommendation system:

1. **Python** – These language are used for overall system backend development, data preprocessing, and algorithm implementation.
2. **HTML/CSS/JAVASCRIPT** - These language are used for frontend development of a system.

**DATASETS:** We have extract a secondary dataset from Kaggle Secondary Datasets which are tmdb\_5000\_movies, tmdb\_5000\_credits.

**5.1.2.** **IMPLEMENTATION DETAILS OF MODULES**

Module used in our system are given as below:

1. **Flask Module**:

The module provides the Flask framework for building the web application.Main component of this module are Flask and request.

1. **Requests Module**:

This module handles HTTP requests to external APIs and also simplify to work with HTTP. It enables website to send and receive the data from website using GET and POST method.

1. **Pandas Module**:

This is a open sourced module which is widely used while working with dataset.It provides data structures and data analysis tools for handling movie data.

1. **Pickle Module**:

This module helps to serializes and deserializes Python objects which is used for loading pickled movie data and similarity matrix. This module make use of component like pickle.

## 5.2 TESTING

System testing is done to examine the artifacts and the behaviour of the software under test by verification and validation. Although each test had a different purpose, each are done to verify that the system elements are properly integrated and is performing the allocated functions. The testing process is done to ensure that the system is meeting the needs and requirements and is performing what it is supposed to be. During testing following goals are tried to achieve:

* To ensure the quality of project.
* To find and eliminate the errors.
* To provide reliability of the system.
* To ensure that the system perform as expected.

## TESTING METHODOLOGIES

There are different testing methodologies used for testing purposes which are as follows:

## 5.2.1.TEST CASES FOR UNIT TESTING

Unit testing is the process of testing the smallest functional unit of code. It is a practice of writing software as a small functional units and then write unit test for each code unit testing is to be performed by developers themselves unit testing can be performed manually but automating the process will speed up the delivery cycle and expand test Coverage. Unit testing will also make debugging easier because it will help to find errors at earlier stage and finding error at earlier stage will help developers to fix them earlier and take less time.

Table 5.1 : Test cases for interface

| S.N |  | Test  cases |  | Features |  | Test Descriptions | Steps to  Execute |  | Expected Results |  | Remarks |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |
| 1.) |  | TC-If-1 |  | User interface |  | Check all the testboxes and buttons | Check pages |  | 1.) UI should be user favourable  2.) Text boxes and buttons should be aligned |  | pass |
|  |  |  |  |  |  |  |  |  |  |  |  |
| 2.) |  | TC-IV-2 |  | Required Fields |  | Check the required fields by not filing any data | Enter invalid movie name |  | Error message should be shown |  | pass |
|  |  |  |  |  |  |  |  |  |  |  |  |
| 3.) |  | TC-IV-3 |  | Auto complete |  | Check whether the movies are shown if only one word is entered | Enter first word  of the movie |  | Movie containing the word entered by the user should be shown |  | pass |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 5.2 : Test Cases for Model Accuracy | | | | | | | | |
|  | S.N |  | Test Cases | Features |  | Test Description | Steps to Execute | Expected Results | Remarks | |
|  |  |  |  |  |  |  |  |  | |
|  | 1.) |  | TC-MA-1 | Recommendation  Testing |  | Tested on test data of our dataset | Check accuracy | Good Accuracy | pass | |

Table 5.3 Test Cases for Performance

| S.N | Test Cases | Features |  | Test Description | Steps to Execute |  | Expected Results | Remarks |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
| 1.) | TC-P-1 | Waiting period |  | System performs efficiently | Check loading time |  | Recommend movie in average time | pass |

## 5.2.2. TEST CASE SYSTEM TESTING

### TEST CASE NUMBER 1

### This test case primarily focuses on verifying that the movie recommendation system recommends movies based solely on cosine similarity.

To evaluate the performance of the content-based movie recommendation system by calculating precision@k, where k is the number of recommendations. For this we have to follow following steps.:

1. **Define Relevant Movies** :In our case, genre, director, cast & crew and overview makes a movie relevant to the user's input.
2. **Generate Recommendations**: When the user inputs their preferred movie, system generate 8 recommendations based on content-based recommendation algorithm.
3. **Manually Evaluate Recommendations**: In this step, we manually review each of the 8 recommended movies and determine how many of them are relevant to the user's input.
4. **Calculate Precision@k**: Use the formula:

In our case: Precision@8 = 6 / 8 = 0.75

This means that out of the 8 recommendations provided, 3 were relevant to the user's input. The precision of our system at k=8 is 0.75.

1. **Interpretation**: A precision score closer to 1 indicates a higher level of precision, meaning that a larger proportion of the recommended items are relevant to the user's preferences. In oour case, a precision score of 0.75 suggests that our recommendation system have higher level of precision.

### TEST CASE NUMBER 2

### This test case primarily focuses whether all the system’s individual component are working together for common objective.

Table 5.4 : System testing

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S.N. | Test Case  ID | Features | Test Case  Description | Steps To  Execute | Expected  ResultS | Remarks |
| 1 | AC-T--1 | Accuracy | Enter prefered movie name and press enter | Check the preference of recommended movie | Most movie are of user’s preference | Pass |
| 2 | AC-T-2 | Data integrity | After movie are recommended click on the respective movie to view its detail | Check whether all the detail regarding movie are correctly provided | All the movies details are correctly provided ensuring data integrity | Pass |

**5.3. RESULT ANALYSIS**

When user opens our system,then he will be asked for his preferred movie as input.When user submit his input to the system then input is further processed by system to calculate the most similar movie on the basis of cosine similarity.Flask’s python code is basically responsible for fetching the most similar movie.After this, it also fetches the detail of movies like poster,release dates,genres and overviews by the help of TMDB API.The Flask application then renders a template with the list of recommended movie titles, posters, release dates, genres, and overviews. It passes this information to the template using the *render\_template()* function that is responsible to display output to the user.An user can then view the recommended movies on the web page and interact with them by clicking on a movie to view more details related to that particular movie.

Content based movie recommendation system performed well as expected. The precision of recommendation provided by the proposed system was found to be 0.75.Recommended movies were exposed to variety of genre, themes and content. Also our system was mainly designed to tackle the challenging problem of cold start. This system was able to accurately recommend movies to new user. Also this system properly handles the increasing workload and growth in number of user.

# CHAPTER 6

# CONCLUSION AND FUTURE RECOMMENDATION

## CONCLUSION

By recommending movies with comparable key properties, such as IMDb votes, average IMDb rating, genre, release year, casts, directors, users, tags, etc. A Recommender systems that leverages content-based filtering and the cosine similarity algorithm may provide users with more insightful and personalized suggestions. These days, Recommender systems are everywhere. They are used by people to locate romantic partners, travel destinations, smart phones, music, and books. Business Insider reports state that more than 80 per cent of Netflix’s business comes from recommendations or suggestions; less than 20 per cent originates from traditional search methods [[11](#Rav21)] .There are Recommenders for almost every kind of information, service, and product available to assist consumers in narrowing down their options to the ones they would find most appealing. . In e-commerce setting, recommender systems enhance revenues, for the fact that they are effective means of selling more products. In scientific libraries, recommender systems support users by allowing them to move beyond catalog searches. Therefore, the need to use efficient and accurate recommendation techniques within a system that will provide relevant and dependable recommendations for users cannot be over-emp [[7](#Con03)]hasized. [[12](#Isi15)]. Recommendation systems are widely used in today’s era of Web 2.0 for searching for reliable and relevant information. While simple recommendation systems recommend users based on few parameters, complex ones take many parameters into consideration. By implementing machine learning in recommender system, intelligent recommendations can be made for customers. Given the potential of such systems, they have a huge commercial value.

## FUTURE RECOMMENDATION

The proposed approach includes content-based filtering, hybrid filtering (combining content and collaborative filtering), and user registration/login for database storage. The recommendation system has the potential to revolutionize the film industry by providing valuable insights and predictions. With advancements in machine learning and data analytics, stakeholders can make more informed decisions. Personalized suggestions can accommodate diverse tastes, and ongoing research can address scalability and data issues. Collaboration with experts can enhance the system's reliability. Joint ventures with streaming services could increase accessibility.

The recommendation system may potentially be used in the future to provide extensive and customized content suggestions for other entertainment industries, such as TV shows and music. It can be employed for diverse set of item recommendation such as books, songs, movies, restaurants, gadgets, e-learning materials etc [[13](#Agg20)]

# REFERENCES

x

|  |  |
| --- | --- |
| [1] | R. N. Behera and S. Dash, "A particle swarm optimization based hybrid recommendation system," *International Journal of Knowledge Discovery in Bioinformatics*, vol. 6, no. 2, pp. 1–10, 2016. |
| [2] | M. Pazzani and D. Billsus, "Content-Based Recommendation Systems," *IEEE Intelligent Systems*, vol. 13, no. 3, pp. 36-41, May/June 1998. |
| [3] | Ken, Theresa Roeder, Dhruv Gupta, and Chris Perkins Goldberg, "Eigentaste: A constant time collaborative filtering algorithm," *Information Retrieval*, vol. 4, pp. 133-151, Jul 2001. |
| [4] | S. Maurya, T. Tripathi, T. Narula, and G. Srivastav R. H. Singh, "Movie Recommendation System using Cosine Similarity and KNN," *International Journal of Engineering and Advanced Technology (Int J Eng Adv Technol)*, vol. 9, no. 5, pp. 556–559, June 2020. |
| [5] | Deepjyoti, and Mala Dutta. Roy, "A systematic review and research perspective on recommender systems.," *Journal of Big Data*, vol. 9, no. 1, p. 59, May 2022. |
| [6] | S. Maurya, T. Tripathi, T. Narula, and G. Srivastav R. H. Singh, "ovie Recommendation System using Cosine Similarity and KNN," *Int J Eng Adv Techno*, vol. 9, no. 5, pp. 556–559, Jun 2020. |
| [7] | Constantinos J. Stefanou, "System Development Life Cycle," *Encyclopedia of Information Systems*, pp. 329-344, 2003. |
| [8] | Uldis Donins Janis Osis, "Unified Modeling Language: A Standard for Designing a Software," *Topological UML Modeling*, pp. 3-51, 2017. |
| [9] | Donald Bell, "UML's sequence diagram.," *IBM.[Online] IBM*, vol. 16, no. 2, Feb 2004. |
| [10] | T. M. Khoshgoftaar X. Su, "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence*, 2009. |
| [11] | Aijaz A. Shaikh and Eldon Li Ravi S. Sharma, "Designing Recommendation or Suggestion Systems: looking to the future," *Electronic Markets*, vol. 31, pp. 243–252, May 2021. |
| [12] | Folasade Olubusola, Yetunde O. Folajimi, and Bolande Adefowoke Ojokoh Isinkaye, "Recommendation systems: Principles, methods and evaluation," *Egyptian Informatics Journal*, vol. 16, no. 3, pp. 261-273, November 2015. |
| [13] | S., Goswami, D., Hooda, M., Chakravarty, A., Kar, A. and Vasudha Aggarwal, "Recommendation systems for interactive multimedia entertainment," *Data Visualization and Knowledge Engineering: Spotting Data Points with Artificial Intelligence*, pp. 23-48, 2020. |

x

# **APPENDICES**

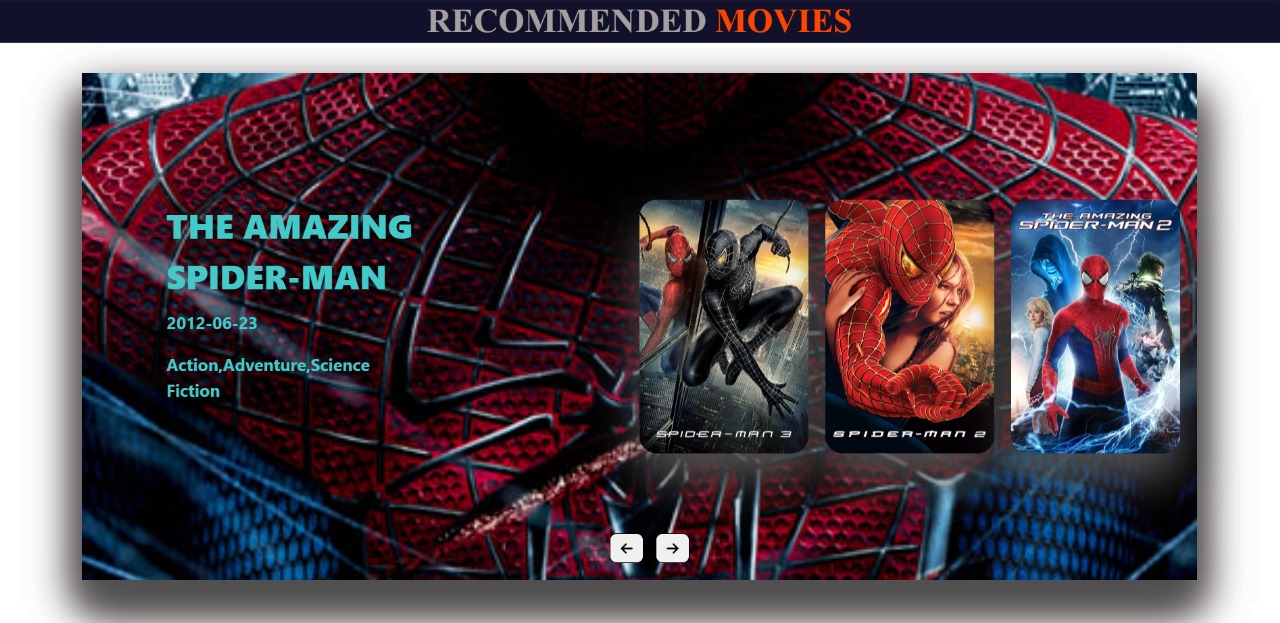
APPENDIX A **:** HOMEPAGE OF PROPOSED SYSTEM

****

APPENDIX B : USER GIVING INPUT

****

APPENDIX C : RECOMMENDATION TO USER

****

APPENDIX D : DETAIL FOR THE MOVIE USER CLICKED FOR

****

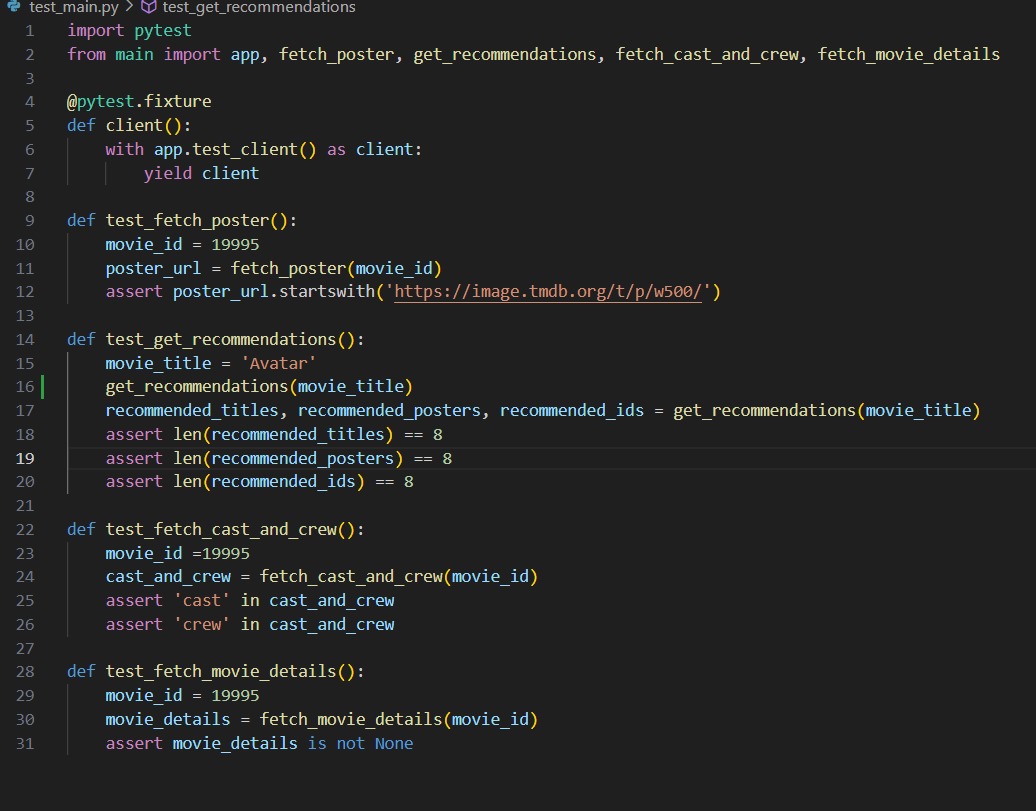
APPENDIX E : DETAIL FOR CREW

****

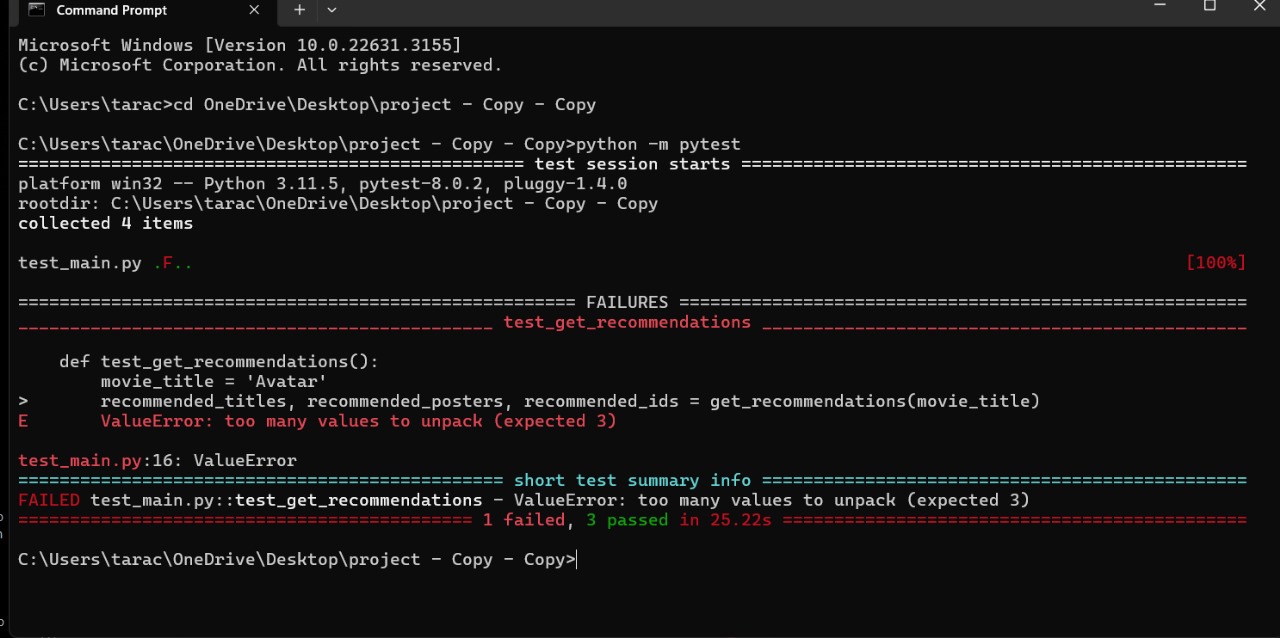
APPENDIX F : DETAILS FOR CAST

****

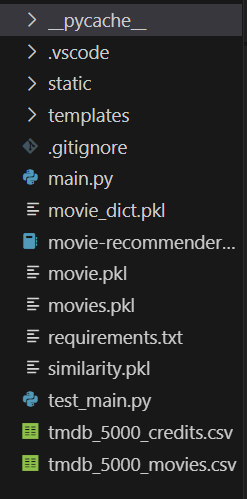
APPENDIX G : CODE FOR UNIT TESTING

****

APPENDIX H : RESULT OF UNIT TESTING

****

APPENDIX I : STRUCTURE OF FLASK FILE

****