PROJECT REPORT

ON

Movie Recommendation System

Submitted

In Partial Fulfilment of

MASTER OF COMPUTER APPLICATIONS (MCA)

Submitted by:
Name – Bhushan Mudliyar
(Roll No-21/FCA/MCA/013)

Under the Supervision of:

(Mr. Manpreet Singh, Assistant Professor)

Faculty of Computer Applications
Manav Rachna International Institute of Research and Studies
(DEEMED TO BE UNIVERSITY)

Sector-43, Aravalli Hills
Faridabad – 121001

June 2023

DECLARATION

I do hereby declare that this project work entitled "**Movie Recommendation System**" submitted by us for the partial fulfilment of the requirement for the award of **MASTER OF COMPUTER APPLICATION MCA** is a record of my work. The report embodies the finding based on my study and observation and has not been submitted earlier for the award of any degree or diploma to any Institute or University.

Signature of Student
Name: Bhushan Mudliyar
Roll No: 21/FCA/MCA/013
Date:

CERTIFICATE FROM THE GUIDE

This is to certify that the project report entitled “**Movie Recommendation System**” submitted in partial fulfilment of the degree of **MASTER OF COMPUTER APPLICATIONS (MCA)** to Manav Rachna International Institute of Research & Studies, Faridabad is an authentic and original work carried out by Bhushan Mudliyar with Roll No. 21/FCA/MCA/013 under my guidance.

Signature of the Guide

Mr. Manpreet Singh,
Assistant Professor
Date:

Head of Department
Dr. Suhail Javed Quraishi
Date:

ACKNOWLEDGEMENT

I gratefully acknowledge for the assistance, cooperation, guidance and clarification provided by **Mr. Manpreet Singh** during the development of **Movie Recommendation System** My extreme gratitude to guided us throughout the project. Without his willing disposition, spirit accommodation, frankness, timely clarification and, above all faith in us, this project could not have been completed in due time. His readiness to discuss all important matters at work deserves special attention of would also like to thank all the faculty members of the computer application department for their cooperation and support.

I would like to extend my sincere gratitude to **Dr. Suhail Javed Quraishi – HoD** for his valuable teaching and advice. I would again like to thank all faculty members of the department for their cooperation and support. I would like to thank the non-teaching staff of the department for their cooperation and support.

I perceive this opportunity as a big milestone in my career development. I will strive to use gained skills and knowledge in the best possible way, and I will continue to work on their improvement, in total in desired career objectives. Hope to continue cooperation with all of you in the future.

Table of Contents

CHAPTER 1- INTRODUCTION

1.1 ABOUT THE ORGANIZATION

In today's digital era, movie watching has become an integral part of our lives. With the increase in the number of streaming services and platforms, the options available to the users have increased manifold. However, with so many options available, it can be overwhelming for users to select a movie to watch. This is where a movie recommendation system comes into play.

Movie recommendation systems are built to help users discover movies that they are likely to enjoy based on their preferences and past viewing behavior. With the vast amount of movies available, it can be overwhelming for users to find movies that align with their interests. Recommendation systems use algorithms and data analysis techniques to provide personalized movie recommendations, making it easier for users to find movies that they are likely to enjoy.

Movie recommendation systems can also benefit content providers, such as streaming platforms and movie rental services, by increasing user engagement and retention. When users are able to find movies that they enjoy more easily, they are more likely to continue using the service and may even recommend it to others.

There are different types of recommendation systems, including collaborative filtering, content-based filtering, and hybrid recommendation systems. Collaborative filtering recommends movies based on the viewing behavior and preferences of similar users, while content-based filtering recommends movies based on the attributes of the movies themselves, such as genre, cast, and plot. Hybrid recommendation systems combine both approaches to provide more accurate and diverse recommendations.

The aim of this project is to develop a content-based movie recommendation system using Python and Machine Learning techniques. Content-based recommendation systems make recommendations based on the similarity between the content of the items. In the case of movies, the system will recommend movies based on the similarity between the features of the movies.

The system will take in input from the user, which will consist of either a movie title or a set of movie features such as genre, director, actor, and release year. The system will then use this information to recommend a list of movies that are similar in content to the input provided by the user.

To achieve this, the project will make use of Python libraries such as pandas, numpy, and scikit-learn. The dataset used for training and testing the model will be obtained from publicly available sources such as tmdb. The system will make use of machine learning algorithms such as K-nearest neighbours (KNN) and Cosine Similarity to identify the movies that are most similar to the user's input.

1.2 **AIMS AND OBJECTIVE**

Aims:

The aim of this project is to develop a content-based movie recommendation system that will assist users in selecting movies to watch. The system will leverage Python and Machine Learning techniques to provide accurate and personalized movie recommendations to the users.

It will improve the user experience by providing personalized movie recommendations based on their viewing behaviour and preferences. By doing so, users can more easily find movies that they are likely to enjoy, increasing their engagement and satisfaction with the content provider's service.

Additionally, movie recommendation systems aim to improve the content provider's business by increasing user engagement and retention. When users are more satisfied with the service, they are more likely to continue using it and may even recommend it to others.

Ultimately, the goal of movie recommendation systems is to create a win-win situation for both the users and the content provider, where users are provided with personalized recommendations and the content provider is able to increase user engagement and satisfaction.

Objectives:

Collect and pre-process a dataset of movies: The project will collect a dataset of movies from publicly available sources such as TMDb and Rotten Tomatoes. The dataset will be pre-processed to ensure data consistency and accuracy.

Feature Engineering: The project will identify the key features that contribute to the similarity between movies. These features may include genre, director, actor, and release year.

Develop a recommendation algorithm: The project will use machine learning algorithms such as K-nearest neighbours (KNN) and Cosine Similarity to develop a recommendation algorithm. The algorithm will take in input from the user and recommend a list of movies that are similar in content to the user's input.

Develop an interactive user interface: The project will provide an interactive user interface where users can input their preferences and receive a list of recommended movies. The interface will be intuitive and user-friendly.

Evaluate the system's accuracy: The project will evaluate the system's accuracy using metrics such as precision, recall, and F1-score. The accuracy will be calculated by comparing the recommended movies to the user's actual viewing history.

Optimize the recommendation algorithm: The project will optimize the recommendation algorithm to improve its accuracy in providing personalized recommendations to the users. This will involve fine-tuning the hyper parameters of the machine learning algorithm.

Deploy the system: The project will deploy the recommendation system to a production environment, making it accessible to users. The system will be scalable and reliable, ensuring that it can handle a large number of user requests.

1.3 MAN POWER

In the case of the movie recommendation system project, the human resources required can be categorized into two broad categories: technical and non-technical.

Technical manpower refers to individuals with the requisite technical skills and knowledge required to design, develop, and deploy the recommendation system. Technical manpower is responsible for collecting and pre-processing the dataset, developing the recommendation algorithm, and designing and deploying the user interface. The technical manpower required for this project may include data scientists, machine learning engineers, and software developers. These individuals should have a strong understanding of Python programming, machine learning, and data analysis.

Non-technical manpower refers to individuals with skills and knowledge required for managing and coordinating the project. This includes project managers, business analysts, and quality assurance personnel. The non-technical manpower is responsible for defining the project scope, identifying project risks, and ensuring that the project is delivered within the allocated time and budget.

The project team should be well balanced and include both technical and non-technical personnel. The team should be able to collaborate effectively to ensure that the project objectives are met. The team should also have good communication skills to ensure that the project progress is communicated effectively to stakeholders.

In addition to the in-house project team, it may be necessary to engage external consultants or vendors to provide specialized expertise or services. This may include data providers, machine learning consultants, and software development vendors. The external resources should be selected based on their technical expertise and track record of delivering quality projects.

CHAPTER 2-SYSTEM STUDY

2.1 **PROPOSED SYSTEM WITH ADVANTAGES**

The proposed movie recommendation system is a content-based system that aims to provide personalized and diverse recommendations to users based on the content of the movies. The system will use machine learning algorithms to analyze the features of the movies such as genre, director, actor, and release year to identify similarities between movies and make recommendations based on the user's preferences. The system will also take into account the user's viewing history and feedback to improve the accuracy of recommendations.

ADVANTAGES:

Personalized recommendations: The proposed system will provide personalized recommendations to users based on their preferences, viewing history, and feedback.

Diverse recommendations: The system will use a content-based approach to recommend movies, which will provide a more diverse set of recommendations than collaborative filtering systems.

No cold start problem: The proposed system will be able to make recommendations for new users who have not yet provided any data, by using the content of the movies to make recommendations.

Accuracy: The system will use machine learning algorithms to analyze the features of the movies and make recommendations, which will improve the accuracy of recommendations.

No privacy concerns: The proposed system will not rely on user data, which will eliminate privacy concerns among users.

Scalability: The proposed system can be easily scaled to handle large datasets of movies and users, making it suitable for use in large streaming platforms.

2.2 **EXIXTING SYSTEM WITH LIMITATION**

- Existing movie recommendation systems can be broadly classified into two categories: collaborative filtering and content-based filtering.

- Collaborative filtering systems rely on the preferences and behaviour of users to make recommendations. The system analyses the viewing history and preferences of users and identifies patterns and similarities between users. Based on these patterns, the system makes recommendations to users. Collaborative filtering systems are popular in movie recommendation systems as they provide personalized recommendations to users. However, the effectiveness of these systems depends on the availability of user data. They may also suffer from the "cold start" problem, where it is difficult to make recommendations for new users who have not yet provided any data.

- Content-based filtering systems, on the other hand, make recommendations based on the content of the items being recommended. In the case of movies, the system analyzes the features of the movies such as genre, director, actor, and release year to identify similarities between movies. Content-based systems can provide recommendations even for new users, and they are not dependent on the availability of user data. However, these systems may suffer from a lack of diversity in recommendations, as they may tend to recommend movies with similar features.

LIMITATIONS

- **Limited personalization:** Collaborative filtering systems may struggle to provide personalized recommendations to users with limited viewing history or preferences.

- **Lack of diversity:** Content-based filtering systems may suffer from a lack of diversity in recommendations, as they may tend to recommend movies with similar features.

- **Cold start problem:** Collaborative filtering systems may struggle to make recommendations for new users who have not yet provided any data.

- **Inaccuracy:** Both collaborative and content-based systems may suffer from inaccuracy in their recommendations, especially if the data used for training is incomplete or biased**.**
- **Data privacy concerns:** Collaborative filtering systems rely on user data, which may raise privacy concerns among users.

2.3 **BRIEF OVERVIEW OF THE TECHNOLOGY**:

2.3.1 Front end: Flask & Streamlit

Streamlit is a free and open-source framework to rapidly build and share beautiful machine learning and data science web apps. It is a Python-based library specifically designed for machine learning engineers.

Data scientists or machine learning engineers are not web developers and they're not interested in spending weeks learning to use these frameworks to build web apps.

Instead, they want a tool that is easier to learn and to use, as long as it can display data and collect needed parameters for modeling. Streamlit allows you to create a stunning-looking application with only a few lines of code.

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

Applications that use the Flask framework include Pinterest and LinkedIn.

2.3.2 Back end: Python, Machine Learning

Python is a high-level programming language that is widely used for web development, data analysis, machine learning, and other applications. It is known for its readability and simplicity, making it a popular choice for beginners and experienced programmers alike.

In web development, Python is often used as a backend language to handle server-side logic and database management. Popular Python web frameworks such as Django and Flask provide tools and libraries that make it easier to build scalable and maintainable web applications.

Machine learning is a subset of artificial intelligence that involves developing algorithms that can learn from data and make predictions or decisions based on that data. Python is one of the most popular languages used for machine learning because of its extensive libraries and frameworks such as NumPy, Pandas, Scikit-learn, and TensorFlow.

In the context of building a movie recommendation system, Python and machine learning would be used to analyse the features of movies such as genre, director, and actors to make personalized recommendations to users. The machine learning algorithms would learn from user data, including viewing history and feedback, to improve the accuracy and relevance of recommendations over time.

- ## 2.3.3 Software Requirement (any one)

The software requirements for the project are:

1. Operating System: The project can run on any operating system, including Windows.
2. PyCharm: PyCharm is an integrated development environment used for programming in Python. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems, and supports web development with Django. PyCharm is developed by the Czech company JetBrains..
3. Web Browser: A modern web browser such as Google Chrome required for running and testing the web application.
4. Jupyter Notbook: Project Jupyter is a project to develop open-source software, open standards, and services for interactive computing across multiple programming languages. It was spun off from IPython in 2014 by Fernando Pérez and Brian Granger

CHAPTER 3- FEASIBILITY STUDY

Overview

The purpose of this feasibility study is to evaluate the feasibility of developing a content-based movie recommendation system using Python, Machine Learning, Flask, and Streamlit. The system will be designed to recommend movies to users based on their preferences and past viewing history.

Technical Feasibility

The technical feasibility of developing the system is high. Python, Machine Learning, Flask, and Streamlit are well-established and widely used technologies. There are also many libraries and tools available that can be used to build the system. Additionally, there are many resources and tutorials available online to help developers learn how to use these technologies.

Operational Feasibility

The operational feasibility of the system is also high. The system will be easy to use and navigate, making it accessible to a wide range of users. The system will also be scalable and able to handle a large number of users and movies.

Economic Feasibility

The economic feasibility of the system is dependent on the budget allocated for the project. Python, Machine Learning, Flask, and Streamlit are open-source technologies, which means that there are no licensing costs associated with their use. However, the cost of hosting the system and storing the movie data will need to be considered.

Legal Feasibility

The legal feasibility of the system is also an important consideration. The system must comply with data privacy and protection laws, such as the General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA). The system must also comply with copyright laws and ensure that it does not infringe on the intellectual property rights of movie studios and distributors.

3.1 **FEASIBILITY STUDY PROCESS**

A feasibility study comprises the following steps.

The feasibility study process involves a systematic and structured analysis of the proposed project to determine whether it is viable, feasible, and desirable. Here are the key steps in the feasibility study process:

Define the Project Scope and Objectives: The first step is to define the project's scope, objectives, and requirements. This includes identifying the problem or opportunity the project aims to address, the stakeholders involved, and the expected outcomes.

Conduct a Preliminary Analysis: The preliminary analysis involves gathering and analyzing data related to the project, such as market research, industry trends, competitor analysis, and financial projections. This analysis helps to identify the project's strengths, weaknesses, opportunities, and threats.

Identify Alternatives: In this step, you identify and evaluate various alternatives or solutions to the problem or opportunity. This may involve considering different technologies, business models, project management approaches, or organizational structures.

Evaluate Alternatives: Once you have identified the alternatives, you need to evaluate each one in terms of its feasibility, cost, benefits, and risks. This analysis helps to determine which alternative is the most viable and suitable for the project.

Develop a Plan: Based on the results of the feasibility study, you need to develop a plan for the project, which includes a detailed project description, scope, schedule, budget, and resources needed. This plan will serve as a guide for the project team and stakeholders.

Present the Findings: Finally, you need to present the findings of the feasibility study to the stakeholders and decision-makers. This presentation should include a summary of the project's objectives, the analysis of the alternatives, the recommended solution, and the project plan.

3.2 **IMPORTANCE OF FEASIBILITY STUDY**

The feasibility study is an essential step in any project's planning process as it helps to evaluate the project's viability, feasibility, and potential risks. Here are some key reasons why the feasibility study is important:

Determine Viability: The feasibility study helps to determine whether the project is viable or not by assessing its technical, operational, financial, legal, and market feasibility. If the study shows that the project is not feasible, it can save the organization time, effort, and resources that would otherwise have been wasted.

Identify Risks: The feasibility study helps to identify potential risks and challenges associated with the project. This enables project managers to develop risk mitigation strategies and contingency plans to minimize or eliminate these risks.

Optimize Resource Allocation: The feasibility study helps to determine the project's resource requirements, such as staff, equipment, and funding. This enables project managers to allocate resources more efficiently and effectively, reducing waste and maximizing project outcomes.

Facilitate Decision-making: The feasibility study provides stakeholders with the necessary information to make informed decisions about the project. This includes evaluating the project's costs, benefits, and risks, and assessing its alignment with organizational goals and priorities.

Enhance Communication and Collaboration: The feasibility study facilitates communication and collaboration between project stakeholders, including project sponsors, team members, and external stakeholders. This enables everyone to understand the project's objectives, requirements, and expectations, and work together to achieve the project's goals.

3.3 **TYPE OF FEASIBILITY STUDY**

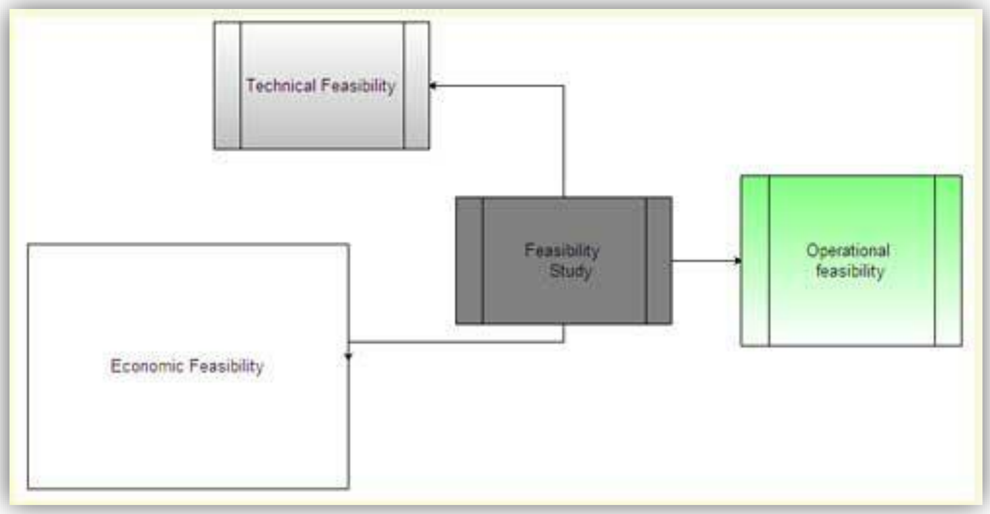


Fig. 1 Feasibility Study

3.3.1 **ECONOMIC FEASIBILITY**:

Economic feasibility is a critical aspect of the feasibility study that focuses on evaluating whether a proposed project is financially viable and sustainable. The economic feasibility study involves assessing the potential costs and benefits of the project, and determining whether the project can generate sufficient returns to justify the investment.

The economic feasibility study typically includes the following key elements:

Project Costs: This involves identifying all the costs associated with the project, including capital expenditures, operating expenses, and other costs such as marketing and advertising. These costs are then analyzed to determine the project's total cost of ownership.

Revenue Projections: This involves estimating the potential revenue that the project can generate. Revenue projections are typically based on market research, industry trends, and sales data.

Cash Flow Analysis: This involves analyzing the project's cash inflows and outflows over the project's lifecycle. The cash flow analysis helps to determine the project's net present value (NPV), internal rate of return (IRR), and payback period.

Sensitivity Analysis: This involves assessing the impact of changes in key variables, such as sales volume, pricing, and costs, on the project's financial performance. Sensitivity analysis helps to identify potential risks and uncertainties that could affect the project's viability.

Risk Analysis: This involves identifying and analyzing the risks associated with the project, such as market risks, financial risks, and operational risks. Risk analysis helps to develop risk management strategies and contingency plans to mitigate the potential impact of these risks.

- ### 3.3.1.1 Possible questions raised in economic analysis are:

- Is the system cost effective?

Ans: Yes, the proposed system is cost-effective in many ways

1. From the patient perspective, it is cost effective.

- The cost of doing full system study.

Ans: Yes, the cost of doing full system study is minimal.

- The cost of business employee time

Ans: Yes, the cost of business employee time is minimal.

- Estimated cost of hardware

Ans: The cost of Hardware is also minimal

- Estimated cost of software/software development

Ans: The cost of software is also minimal

Hence, the proposed system is economically feasible.

3.3.2 **BEHAVIORAL FEASIBILITY**:

Behavioral feasibility is a critical aspect of the feasibility study that focuses on evaluating whether a proposed project is acceptable and feasible from a human and organizational perspective. The behavioral feasibility study involves assessing the project's impact on people, including users, employees, and other stakeholders, and determining whether they are likely to adopt and support the project.

The behavioral feasibility study typically includes the following key elements:

User Acceptance: This involves assessing the project's potential impact on users, including their willingness to adopt the new system, their ability to use it, and their overall satisfaction with it. User acceptance testing can help to identify any usability issues or design flaws that need to be addressed before the project is launched.

Organizational Acceptance: This involves assessing the project's impact on the organization, including its impact on workflow, processes, and organizational culture. Organizational acceptance testing can help to identify any potential resistance to change and develop strategies to address it.

Training Needs: This involves identifying the training needs of users and employees to ensure they can use the new system effectively. Training programs can help to increase user adoption and improve the overall success of the project.

Change Management: This involves developing strategies to manage the transition to the new system, including communication plans, stakeholder engagement, and organizational change management. Change management can help to minimize resistance to change and ensure a smooth transition to the new system.

Sustainability: This involves assessing the long-term viability of the project and its ability to meet the changing needs of users and the organization. Sustainability testing can help to identify any potential risks or issues that need to be addressed to ensure the project's long-term success.

3.3.2.1 Possible questions raised in behaviour analysis are:

- **Is the audience likely to adopt the behavior?** Is the current behavior seen as a problem? How engrained or "rewarding" are the current or competing behaviors?

Ans: Yes, the audience is likely to adopt the behavior. Because the proposed system changes the total environment of the previous system.

- **How costly is it** (time, effort, resources) for the audience segment to perform the behavior?

Ans: The Proposed system is made up at a low-cost model in terms of time, effort, and resources.

- **How complex is the behavior** (does it involve few or several elements)?

Ans: The Proposed system is quite easy to use.

Hence, the proposed system is operationally feasible.

3.3.3 **TECHNICAL FEASIBILITY**:

Technical feasibility is a critical aspect of the feasibility study that focuses on evaluating whether a proposed project can be implemented using available technology and resources. The technical feasibility study involves assessing the project's technical requirements, evaluating available technology solutions, and determining whether the project can be implemented using existing technology and infrastructure.

The technical feasibility study typically includes the following key elements:

Technical Requirements: This involves identifying the project's technical requirements, including hardware, software, and network infrastructure. This helps to ensure that the project can be implemented using available technology solutions.

Technology Assessment: This involves assessing available technology solutions, including hardware and software, and evaluating their suitability for the project. This helps to identify any gaps in the technology infrastructure that need to be addressed before the project can be implemented.

Data Requirements: This involves identifying the project's data requirements, including data sources, storage, and processing requirements. This helps to ensure that the project can be implemented using available data resources.

Technical Architecture: This involves designing the technical architecture for the project, including hardware and software components, data storage, and network infrastructure. This helps to ensure that the project can be implemented using a scalable and secure architecture.

Resource Requirements: This involves identifying the resources required to implement the project, including technical staff, hardware, software, and network infrastructure. This helps to ensure that the project can be implemented using available resources.

3.3.3.1 **QUESTIONS THAT IT ANSWERS**

- Is the project feasible within the limits of current technology?

Ans: Yes, the Project is feasible within the limits of current technology

- Does the technology exist at all?

Ans: No

- Is it a practical proposition?

Ans: Yes

- Is there enough manpower- programmers, testers & debuggers?

Ans: Yes

- Do the required software and hardware exist?

Ans: Yes

- Are the current technical resources sufficient for the new system?

Ans: Yes

- Can they be upgraded to provide the level of technology necessary for the new system?

Ans: Yes

- Can the technology be easily applied to current problems?

Ans: Yes

- Does the technology have the capacity to handle the solution?

Ans: Yes

3.3.3.2 **Technical feasibility also performs the following tasks**.

- Analyzes the technical skills and capabilities of the software development team members.

- Determines whether the relevant technology is stable and established.

- Ascertains that the technology chosen for software development has a large number of users so that they can be consulted when problems arise or improvements are required.

Hence, we can say that the proposed system is technically feasible

3.4 HARDWARE AND SOFTWARE SPECIFICATIONS

This chapter involves both the hardware and software requirements needed for the Project and detailed explanation of the specifications.

Hardware:

A PC with Windows/Linux OS

Processor with 1.7-2.4gHz speed

Minimum of 8GB RAM

2GB Graphic card

Software:

Text Editor (VS-code/WebStorm)

Anaconda distribution package (PyCharm Editor)

Python libraries

Anaconda distribution:

Anaconda is a free and open-source distribution of the Python programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management system and deployment. Package versions are managed by the package management system conda. The anaconda distribution includes data-science packages suitable for Windows, Linux and MacOS.3

Python libraries:

For the computation and analysis we need certain python libraries which are used to perform analytics. Packages such as SKlearn, Numpy, pandas, Matplotlib, Flask framework, etc are needed.

SKlearn: It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

NumPy: NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python.

Pandas: Pandas is one of the most widely used python libraries in data science. It provides high-performance, easy to use structures and data analysis tools. Unlike NumPy library which provides objects for multi-dimensional arrays, Pandas provides in-memory 2d table object called Data frame.

Flask: It is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug

CHAPTER 4 –PROJECT MONITORING SYSTEM

Monitoring is the regular observation and recording of activities taking place in a project or program. It is a process of routinely gathering information on all aspects of the project.

To monitor is to check on how project activities are progressing. It is observation; ─ systematic and purposeful observation.

Monitoring also involves giving feedback about the progress of the project to the donors, implementors, and beneficiaries of the project.

Reporting enables the gathered information to be used in making decisions for improving project performance.

Purpose of Monitoring:

Monitoring is very important in project planning and implementation.

It is like watching where you are going while riding a bicycle; you can adjust as you go along and ensure that you are on the right track.

MONITORING PROVIDES INFORMATION THAT WILL BE USEFUL IN:

- Analysing the situation in the community and its project;
- Determining whether the inputs in the project are well utilized;
- Identifying problems facing the community or project and finding solutions;
- Ensuring all activities are carried out properly by the right people and in time;
- Using lessons from one project experience on to another; and
- Determining whether the way the project was planned is the most appropriate way of solving the problem at hand.

4.1 **Gantt Chart**

A Gantt chart is a graphical depiction of a project schedule. It is a type of bar chart that shows the start and finish dates of several elements of a project that including resources, milestones, tasks, and dependencies.

The Gantt chart is the most widely used chart in project management. These charts are useful in planning a project and defining the sequence of tasks that require completion. In most instances, the chart is displayed as a horizontal bar chart. Horizontal bars of different lengths represent the project timeline, which can include task sequences, duration, and the start and end dates for each task. The horizontal bar also shows how much of a task requires completion. The length of the bar is proportional to the time necessary for a task's completion. The project tasks are represented on the vertical axis.

1. Project planning and analysis: 2 weeks

- Defining project requirements
- Identifying project goals and objectives
- Conducting feasibility study
- Creating project schedule

2. Front-end development: 4 weeks

- Designing website layout and user interface
- Creating website mockups and wireframes
- Integrating front-end components with back-end services
- Testing and debugging front-end code

3. Back-end development: 6 weeks

- Designing database schema and data models
- Implementing user authentication and authorization
- Writing server-side scripts and services
- Testing and debugging back-end code

4. Database development: 4 weeks

- Creating collections and indexes
- Implementing database queries and updates
- Testing and debugging database code

5. Integration and testing: 4 weeks

- Integrating front-end and back-end components
- Testing the application for functionality and usability
- Conducting load testing and performance testing
- Fixing bugs and issues

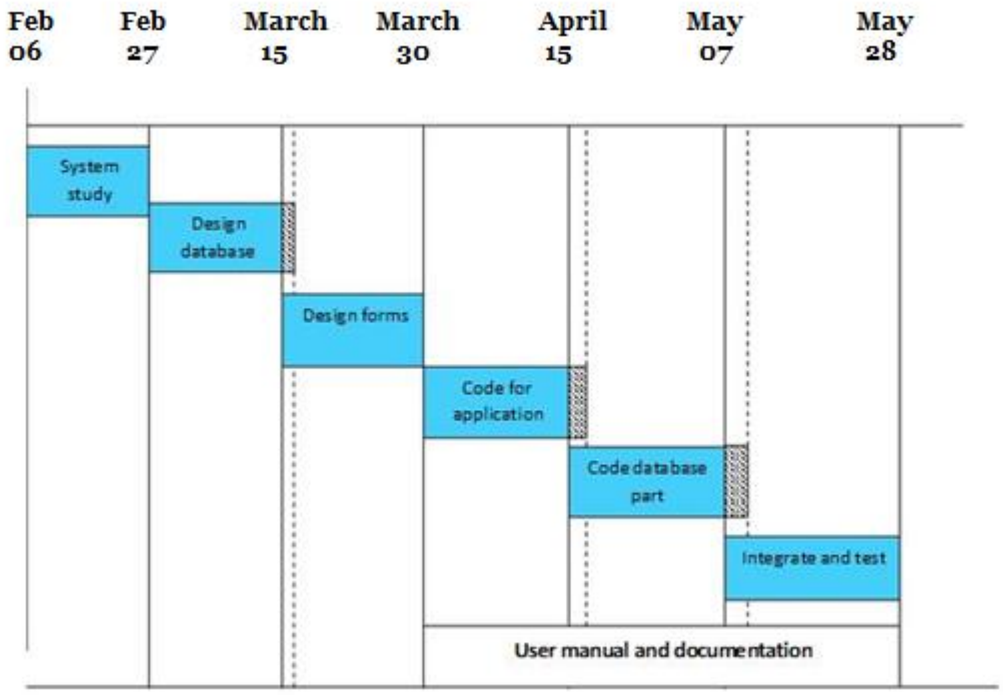


Fig -2 Gantt Chart

- ### 4.1.1 HOW IT WORKS

A Gantt chart helps in scheduling, managing, and monitoring specific tasks and resources in a project. The chart shows the project timeline, which includes scheduled and completed work over a period. The Gantt chart aids project managers in communicating project status or plans and also helps ensure the project remains on track.

The chart identifies tasks that may be executed in parallel and those that cannot be started or finished until other tasks are complete. The Gantt chart can help detect potential bottlenecks and identify tasks that may have been excluded from the project timeline.

The chart depicts task slack time or additional time for completion of a task that should not delay the project, noncritical activities that may be delayed and critical activities that must executed on time.

CHAPTER 5- SYSTEM ANALYSIS

Systems analysis is a problem-solving method that involves looking at the wider system, breaking apart the parts, and figuring out how it works in order to achieve a particular goal. But before we get into detail about how that works, we should probably first answer the question: What is a system?

A **system** is a general set of parts, steps, or components that are connected to form a more complex whole. For example, a computer system contains processors, memory, electrical pathways, a power supply, etc. For a very different example, a business is a system made up of methods, procedures, and routines.

The first step in solving a problem that involves a system is analyzing that system. This involves breaking it down into the parts that make it up, and seeing how those parts work together. Sometimes figuring out how a system works can involve turning off parts of the system and seeing what happens, or changing parts of the system and seeing what the result is. If you change what goes into a system, how does it change what comes out? Basically, systems analysis involves techniques that allow you to understand how a system works.

5.1 **BENEFITS OF SYSTEMS ANALYSIS**

There are many reasons why you might want to analyze a system. These include learning to use systems that somebody else created, for planning new systems, and reducing errors when problem solving.

Sometimes systems analysis is a necessity. For example, if you buy a company and want to hire your own staff, you might find out that you are now an owner of a series of systems you know nothing about. It might be impossible to use the systems until they are analyzed, especially if all of the original staff are gone. This can happen on a much smaller scale when you start a new job, or move to a new department. If there isn't anyone there to explain how a system works, you might have to figure it out for yourself.

Installing a new system, whether it's a home entertainment center, a factory production line, or way of working in an office, requires proper planning. Without this planning and systems analysis, the change might not work. You need to understand how the current system works before you install a new one. Otherwise, there may be problems, and the project could be a failure

.

5.2 REQUIREMENT SPECIFICATIONS

Requirement specifications are the detailed descriptions of the functional and non-functional requirements of a movie recommendation system. These specifications serve as the basis for the development and implementation of the system, and they provide a clear understanding of what the system is expected to do and how it should behave.

Requirement specifications should include the following information:

Purpose and scope: A description of the intended purpose and scope of the system, including its target audience and the problems it is intended to solve.

User requirements: A detailed description of the functional requirements of the system, including the specific features and capabilities that the system should have to meet user needs.

System requirements: A detailed description of the non-functional requirements of the system, including performance, scalability, reliability, usability, compatibility, accessibility, security, and compliance.

Use cases: A series of scenarios that describe how the system will be used, including the specific steps involved and the expected outcomes.

Technical requirements: A detailed description of the technical environment in which the system will operate, including the hardware and software components that will be required, as well as any relevant standards or protocols.

Acceptance criteria: A set of criteria that will be used to determine whether the system meets the requirements and is ready for deployment.

Constraints and assumptions: Any constraints or assumptions that will affect the design or implementation of the system, such as time or budget constraints, or technical limitations.

By documenting the requirements specifications in detail, the development team can ensure that they have a clear understanding of what is required and can work towards meeting those requirements in the design and implementation of the system. Additionally, the specifications can serve as a basis for testing and validation to ensure that the system meets the needs of the users and the organization.

5.2.1 FUNCTIONAL REQUIREMNTS

Functional requirements are the specific features and capabilities that a movie recommendation system must have in order to meet its intended purpose. Some common functional requirements of a movie recommendation system may include:

User profile creation: The system should allow users to create a profile where they can provide information about their preferences, interests, and past viewing behavior.

Movie database: The system should have a comprehensive database of movies with attributes such as genre, cast, director, plot summary, and user ratings.

Recommendation algorithm: The system should have an algorithm that uses the user's profile and movie attributes to generate personalized recommendations.

Filtering and sorting: The system should allow users to filter and sort movie recommendations based on different attributes such as genre, release date, and user rating.

Feedback mechanism: The system should allow users to rate and provide feedback on recommended movies, which can be used to improve the accuracy of future recommendations.

Integration with external services: The system may need to integrate with external services such as streaming platforms, movie rental services, or social media platforms to provide seamless user experience.

Security and privacy: The system should have appropriate security measures in place to protect user data and ensure user privacy.

These are just a few examples of functional requirements that a movie recommendation system may have. The specific requirements will vary depending on the intended purpose, target audience, and technical capabilities of the system.

5.2.2 NON-FUNCTIONAL REQUIREMNTS

Non-functional requirements are the characteristics and constraints that a movie recommendation system must meet, but that are not directly related to its specific features or capabilities. Some common non-functional requirements of a movie recommendation system may include:

Performance: The system should be able to provide recommendations quickly and efficiently, without significant delays or interruptions.

Scalability: The system should be able to handle large amounts of data and traffic as the user base grows.

Reliability: The system should be able to operate continuously and without errors, with minimal downtime or maintenance requirements.

Usability: The system should be user-friendly and easy to navigate, with clear and concise instructions and interfaces.

Compatibility: The system should be compatible with different devices and platforms, such as web browsers, mobile devices, and operating systems.

Accessibility: The system should be accessible to users with different needs and abilities, such as those with visual impairments or limited mobility.

Security: The system should have appropriate security measures in place to protect user data and prevent unauthorized access.

Compliance: The system should comply with relevant laws, regulations, and industry standards, such as data protection regulations and best practices in information security.

These are just a few examples of non-functional requirements that a movie recommendation system may have. The specific requirements will depend on the intended use, target audience, and technical environment of the system.

5.3 **DFDs**

A **Data Flow Diagram** (**DFD**) is a graphical representation of the "flow" of data through an information system. A DFD is often used as a preliminary step to create an overview of the system. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel.

- ## 5.3.1 CONSTRUCTING A DFD:

Several rules of thumb are used in drawing DFD'S:

1. Process should be named and numbered for an easy reference. Each name should be representative of the process.
2. The direction of flow is from top to bottom and from left to right. Data traditionally flow from source to the destination although they may flow back to the source. One way to indicate this is to draw long flow line back to a source. An alternative way is to repeat the source symbol as a destination. Since it is used more than once in the DFD it is marked with a short diagonal.
3. When a process is exploded into lower level details, they are numbered.
4. The names of data stores and destinations are written in capital letters. Process and dataflow names have the first letter of each work capitalized

A DFD typically shows the minimum contents of data store. Each data store should contain all the data elements that flow in and out.

Questionnaires should contain all the data elements that flow in and out. Missing interfaces redundancies and like is then accounted for often through interviews.

- ## 5.3.2 SAILENT FEATURES OF DFD'S

1. The DFD shows flow of data, not of control loops and decision are controlled considerations do not appear on a DFD.
2. The DFD does not indicate the time factor involved in any process whether the dataflow take place daily, weekly, monthly or yearly.
3. The sequence of events is not brought out on the DFD

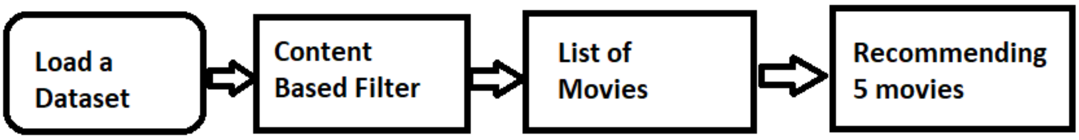


Fig.3 DFD

Initially load the data sets that are required to build a model the data set that are required in this project are movies.csv, ratinfg.csv, users.csv all the data sets are available in the Kaggle.com. Basically, in this project content based produce a list of movies to a particular user by combining both based on the user id a single final list of movies are recommended to the particular user.

CHAPTER-6 SYSTEM DESIGN

System design is the process of designing the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system.

System Analysis is the process that decomposes a system into its component pieces for the purpose of defining how well those components interact to accomplish the set requirements.

The purpose of the System Design process is to provide sufficient detailed data and information about the system and its system elements to enable the implementation consistent with architectural entities as defined in models and views of the system architecture.

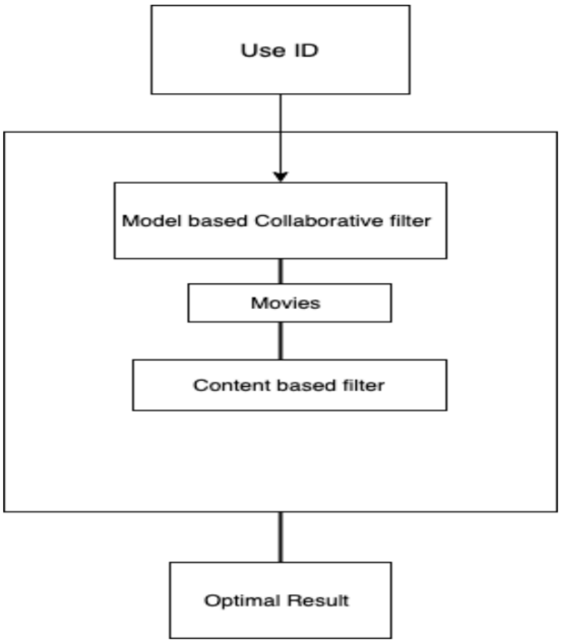
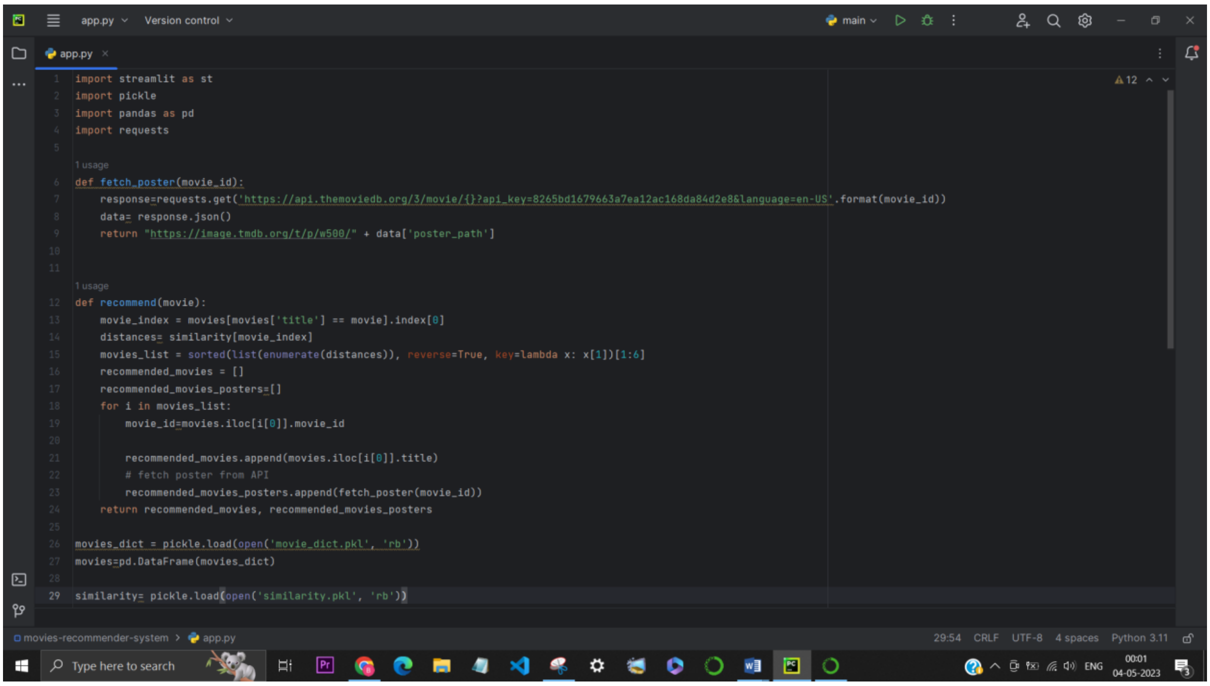


Fig. 4 System Design

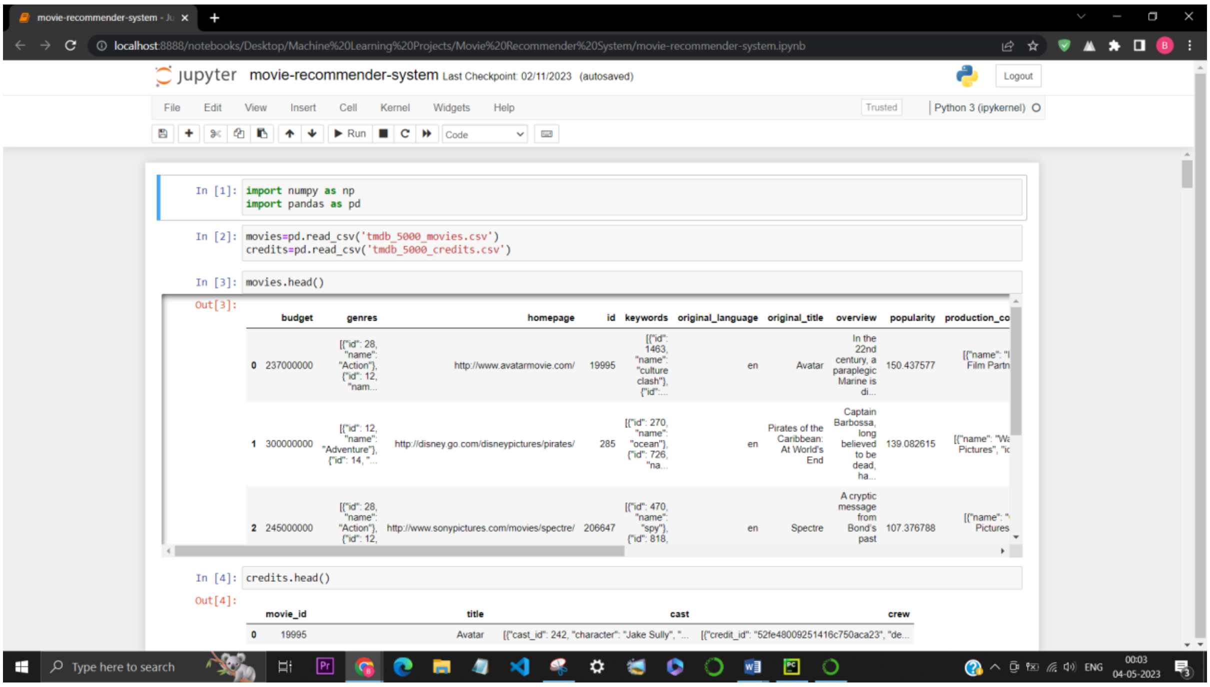
CHAPTER-7   INPUT/OUTPUT FROM DESIGN

Code: Front-end



```python
import streamlit as st
import pickle
import pandas as pd
import requests

def fetch_poster(movie_id):
    response=requests.get('https://api.themoviedb.org/3/movie/{}?api_key=8265bd1679663a7ea12ac168da84d2e8&language=en-US'.format(movie_id))
    data= response.json()
    return "https://image.tmdb.org/t/p/w500/" + data['poster_path']


def recommend(movie):
    movie_index = movies[movies['title'] == movie].index[0]
    distances= similarity[movie_index]
    movies_list = sorted(list(enumerate(distances)), reverse=True, key=lambda x: x[1])[1:6]
    recommended_movies = []
    recommended_movies_posters=[]
    for i in movies_list:
        movie_id=movies.iloc[i[0]].movie_id

        recommended_movies.append(movies.iloc[i[0]].title)
        # fetch poster from API
        recommended_movies_posters.append(fetch_poster(movie_id))
    return recommended_movies, recommended_movies_posters

movies_dict = pickle.load(open('movie_dict.pkl', 'rb'))
movies=pd.DataFrame(movies_dict)

similarity= pickle.load(open('similarity.pkl', 'rb'))
```

Code: Back-end


movie-recommender-system - Ju
localhost:8888/notebooks/Desktop/Machine%20Learning%20Projects/Movie%20Recommender%20System/movie-recommender-system.ipynb
jupyter movie-recommender-system Last Checkpoint: 02/11/2023 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
Type here to search
ENG
00:03
04-05-2023

```
In [1]: import numpy as np
        import pandas as pd
```

```
In [2]: movies=pd.read_csv('tmdb_5000_movies.csv')
        credits=pd.read_csv('tmdb_5000_credits.csv')
```

```
In [3]: movies.head()
```

Out[3]:

	budget	genres	homepage	id	keywords	original_language	original_title	overview	popularity	production_co
0	237000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	http://www.avatarmovie.com/	19995	[{"id": 1463, "name": "culture clash"}, {"id":...	en	Avatar	In the 22nd century, a paraplegic Marine is di...	150.437577	[{"name": "I Film Partn
1	300000000	[{"id": 12, "name": "Adventure"}, {"id": 14, "...	http://disney.go.com/disneypictures/pirates/	285	[{"id": 270, "name": "ocean"}, {"id": 726, "na...	en	Pirates of the Caribbean: At World's End	Captain Barbossa, long believed to be dead, ha...	139.082615	[{"name": "Wa Pictures", "ic
2	245000000	[{"id": 28, "name": "Action"}, {"id": 12,	http://www.sonypictures.com/movies/spectre/	206647	[{"id": 470, "name": "spy"}, {"id": 818,	en	Spectre	A cryptic message from Bond's past	107.376788	[{"name": " Pictures

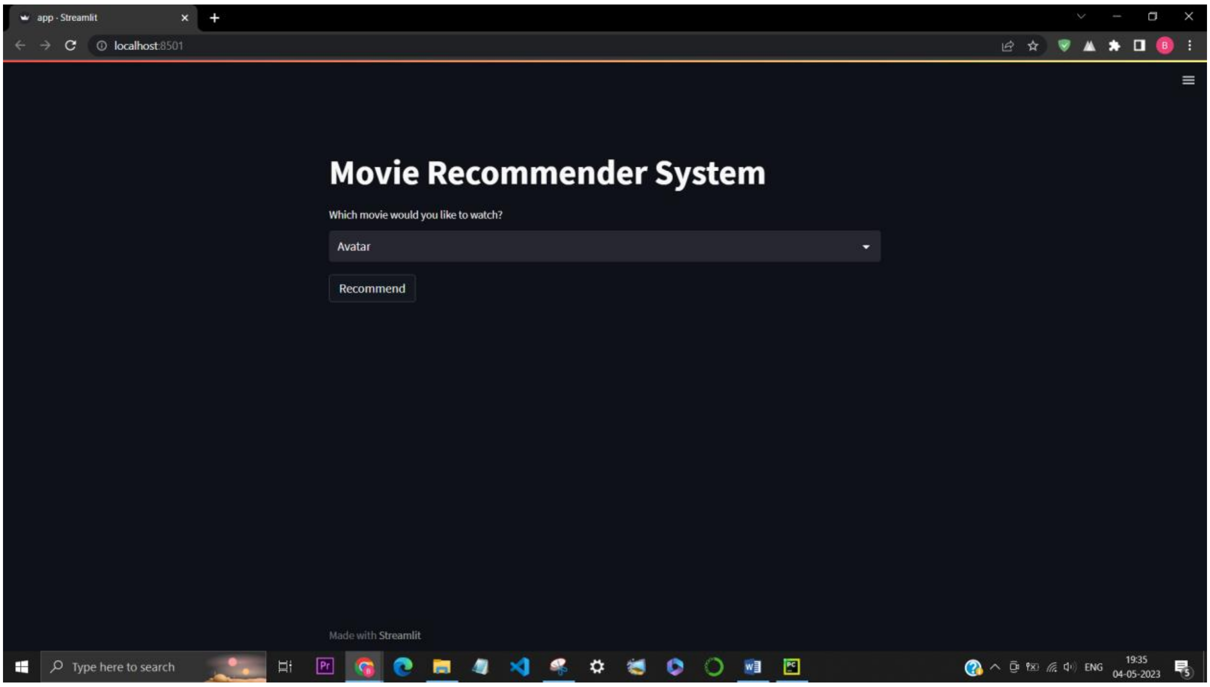
```
In [4]: credits.head()
```

Out[4]:

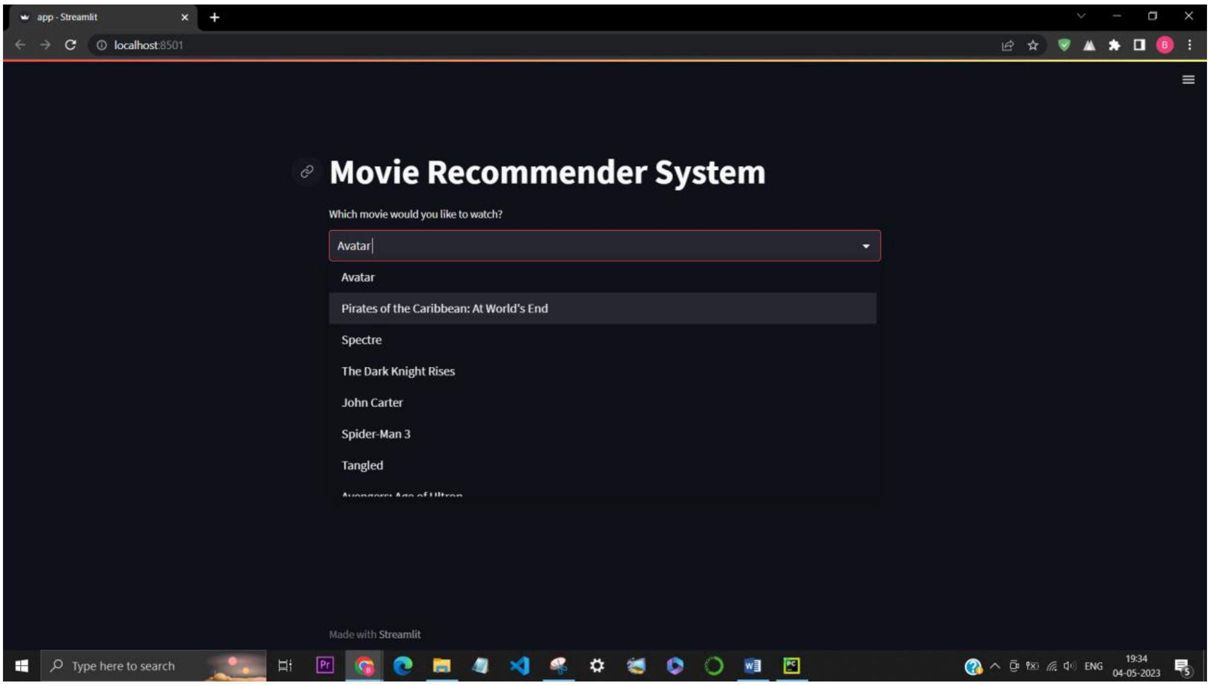
	movie_id	title	cast	crew
0	19995	Avatar	[{"cast_id": 242, "character": "Jake Sully", "...	[{"credit_id": "52fe48009251416c750aca23", "de...

Output:

Selecting movie out of 5000.



List of movies:



Movie Recommender System

Which movie would you like to watch?

Avatar

Avatar

Pirates of the Caribbean: At World's End

Spectre

The Dark Knight Rises

John Carter

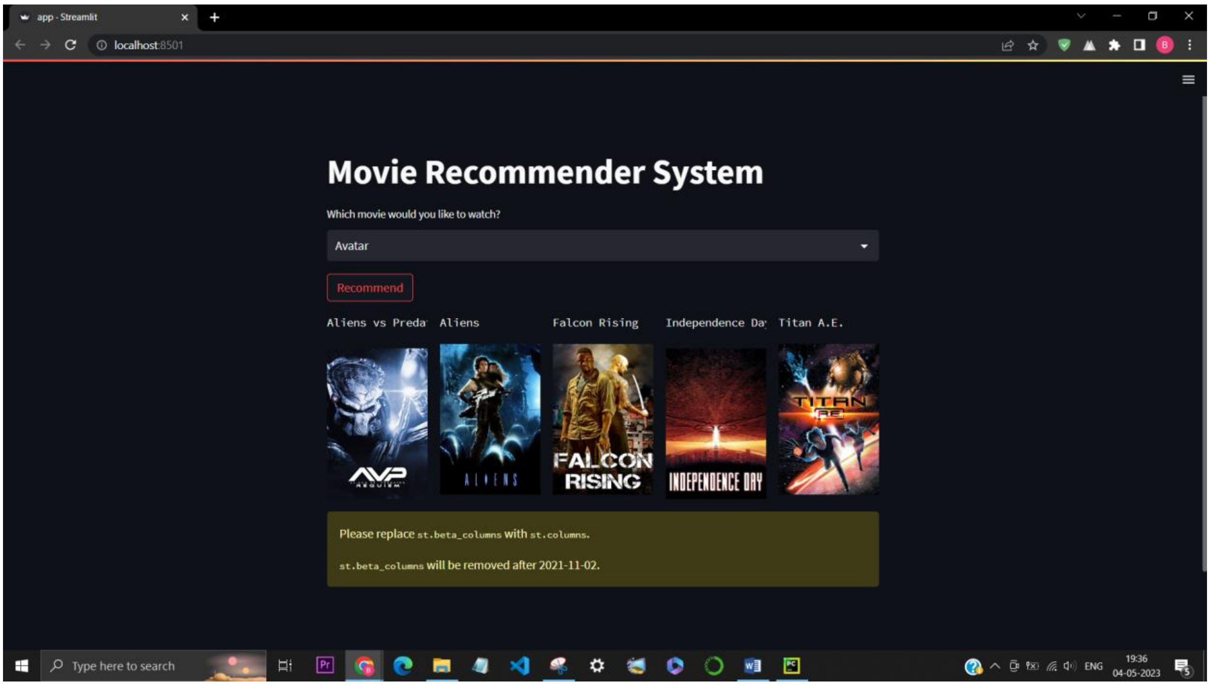
Spider-Man 3

Tangled

Avengers: Age of Ultron

Made with Streamlit

Recommended movies:



Movie Recommender System

Which movie would you like to watch?

Avatar

Aliens vs Preda Aliens Falcon Rising Independence Day Titan A.E.

Please replace `st.beta_columns` with `st.columns`.

`st.beta_columns` will be removed after 2021-11-02.

CHAPTER 8- SYSTEM TESTING

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that all the system elements have been properly integrated and perform allocated functions. The testing process is actually carried out to make sure that the product exactly does the same thing what is supposed to do. In the testing stage following goals are tried to achieve: -

● To affirm the quality of the project.

● To find and eliminate any residual errors from previous stages.

● To validate the software as a solution to the original problem.

● To provide operational reliability of the system.Testing can't show the shortfall of deformities, it can just show that product absconds are available.

- **Unit Testing:** This is the lowest level of testing that is conducted to remove syntax & logic errors from a single unit. Individual components are tested to ensure that they ensure that they operate correctly. The motivation behind unit testing is to distinguish and right whatever number inside rationale mistakes as would be prudent. Unit tests would be repeatable and might be led anytime in the execution interaction as per the supported unit test plan for the module I project the objective for unit testing by engineers is to perform chosen way testing in which each influenced branch is explored in all potential ways at any rate once and each influenced line of code is executed in any event once. The engineer would do unit testing at season of coding.

- **Module testing:** A module is a collection of dependent components such as object class, an abstract data type or some looser collection of procedure and functions. A module encapsulates related components, so can be tested without other system modules.

- **Sub System Testing:** This phase involves testing collections of modules, which have been integrated into sub system. This test for Problem that arise from component interactions. This testing should begin as soon as usable versions of some of the system components are available.

- **System Testing:** The sub-system are integrated to make up the system. The system as a completely entity is tested over here. This process is concerned with finding errors that result from unanticipated interaction between the sub-systems. It is also concerned with validating that the system meets its functional & non functional requirements & testing with emergent system properties.

- **Acceptance Testing:** This is the final stage in the testing process before the system is accepted for operational use. The system is tested with Data supplied by the system customer rather than simulated test data. Acceptance testing may reveal errors & omissions in the system requirements definition because the real data exercise the system in different ways from the test data. It may also reveal requirements problems where the system's facilities do no really meet the user's needs or the system performance in acceptance.

- **Integration Testing-** Integration testing is a procedure used to validate that the modules or units of source code or functions are working properly. Ideally, module is integrated which is independent from the others; mock objects can be used to assist testing a module or a piece of a module in isolation. Integration testing is typically done by the developers and by end-users.

- **Functional Testing-** The goal of this test is to guarantee that every component of the application meets the utilitarian necessities as illustrated in framework determinations. Useful testing covers the parts of the framework executing capacities it should execute- including client orders, information control, searches and business measures, client screens, and reconciliations. Useful testing covers the undeniable surface sort of capacities, just as the back-end activities, (for example, security and what updates mean for the framework). Prior to executing the framework experiments in full, a restricted utilitarian testing will be performed with a subset of framework experiment where the framework will be run on two (or might be more is to be chosen) work days and covering start to finish two (or might be more is to be chosen) occasion types. This is done to confirm if every one of the parts of the framework is introduced appropriately and to do an essential usefulness testing. This will close undeniable level testing. It will be trailed by itemized level tests, which will mean to test the individual cycles and information streams.

8.1 THERE ARE TWO MAJOR TYPE OF TESTING THEY ARE

- **WHITE BOX TESTING**
- **BLACK BOX TESTING**

8.1.1 WHITE BOX TESTING

White Box Testing is characterized as the testing of a product arrangement's inner construction, plan, and coding. In this sort of testing, the code is apparent to the analyzer. It centers fundamentally around checking the progression of data sources and yields through the application, improving plan and ease of use, reinforcing security. White box testing is otherwise called Clear Box testing, Open Box testing, Structural testing, Transparent Box testing, Code-Based testing, and Glass Box testing. It is generally performed by engineers.

White box testing includes the testing of the product code for the accompanying:

- Internal security openings
- Broken or inadequately organized ways in the coding measures
- The stream of explicit contributions through the code
- Expected output
- The usefulness of contingent circles

- ## 8.1.2 BLACK BOX TESTING

Discovery testing is a strategy for programming testing that looks at the usefulness of an application dependent on the particulars. It is otherwise called Specifications based testing. Autonomous Testing Team normally plays out this kind of testing during the product testing life cycle.

This technique for test can be applied to every single degree of programming testing like unit, mix, framework and acknowledgment testing.

- ### 8.1.3 TESTING PRINCIPLES:

- Testing shows presence of deformities
- Exhaustive testing is unimaginable
- Early testing
- Defect grouping
- Pesticide conundrum
- Testing is setting subordinate
- Absence of blunders false notion

Overall, system testing is crucial for ensuring that the PanditJiPot website is functional, reliable, and meets the user requirements. Different types of testing can be conducted to identify and resolve any issues or bugs before the website is launched.

CHAPTER 9- SYSTEM IMPLEMENTATION

9.1 System Requirement

The Proposed System Make Use Different Algorithms and Methods

Cosine Similarity: Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them.

The cosine of two non-zero vectors can be derived by using the Euclidean dot product formula:

$$\mathbf{A} \cdot \mathbf{B} = \|\mathbf{A}\| \, \|\mathbf{B}\| \cos\theta$$

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

Cosine similarity is a widely used technique in movie recommendation systems to measure the similarity between two movies based on their attributes. It is a similarity measure that determines the cosine of the angle between two vectors in a high-dimensional space.

In a movie recommendation system, each movie is represented as a vector of numerical features such as the movie genre, cast, director, plot summary, and user ratings. The cosine similarity between two movies is calculated by taking the dot product of their feature vectors and dividing it by the product of their vector magnitudes. This results in a value between -1 and 1, where a value of 1 indicates that the two movies are identical and a value of 0 indicates that they are completely dissimilar.

Using cosine similarity, a recommendation system can identify movies that are similar to a user's preferred movies. For example, if a user has previously watched and enjoyed action movies, the system can recommend other action movies with high cosine similarity scores. This can help the system provide personalized recommendations that align with the user's preferences.

Overall, cosine similarity is a useful tool in movie recommendation systems as it can help identify similarities between movies based on their features, allowing for more accurate and personalized recommendations.

DOCUMENTATION

Documentation is an important aspect of any software project, and it helps in providing information about the project and its features to users and developers. The documentation for the PanditJiPot project can include the following:

1. User manual: A user manual is a document that provides instructions on how to use the website, including how to navigate through the different pages and how to perform various actions, such as searching for products, adding items to cart, and making payments.
2. Technical documentation: Technical documentation is a detailed document that provides information on the project architecture, data models, database schema, APIs, and other technical aspects of the project. This documentation helps developers understand the system and its components, and it can be used for troubleshooting, maintenance, and future development.
3. Installation guide: An installation guide is a document that provides step-by-step instructions for installing and configuring the project on a server. It includes information on the software requirements, configuration settings, and any dependencies that need to be installed.
4. Release notes: Release notes are a document that provides information on the changes made to the system in each release. It includes details on new features, bug fixes, and any known issues.
5. Project plan: A project plan is a document that outlines the project timeline, milestones, and deliverables. It helps to track progress and ensure that the project is on schedule.
6. Requirements document: A requirements document is a document that outlines the project requirements, including the functional and non-functional requirements, use cases, and user stories.
7. Design document: A design document is a document that provides a high-level overview of the system architecture, design patterns, and other design decisions.

Documentation is an ongoing process and should be updated regularly to reflect changes to the project.

SCOPE OF THE PROJECT

In the proposed approach, It has considered Genres of movies but, in future we can also consider age of user as according to the age movie preferences also changes, like for example, during our childhood we like animated movies more as compared to other movies.

There is a need to work on the memory requirements of the proposed approach in the future. The proposed approach has been implemented here on different movie datasets only. It can also be implemented on the Film Affinity and Netflix datasets and the performance can be computed in the future.

We can use different hybrid approach which is nowadays used in social media like Instagram, YouTube for better content suggestion.

BIBLIOGRAPHY

1. **https://www.google.com/search?q=w3school&oq=w3school&aqs=chrome..69i57.5824j0j 7&sourceid=chrome&ie=UTF-8**

2. **https://www.google.com/search?q=open+cart&oq=open+cart+&aqs=chrome..69i57.5190 j0j9&sourceid=chrome&ie=UTF-8**

3. **https://www.google.com/search?q=youtube&oq=yotu&aqs=chrome.1.69i57j0l5.2349j0j9 &sourceid=chrome&ie=UTF-8**