GLA University, Mathura

****

**Topic**

Mini Project Synopsis on

Maverick: The Movie Recommendation System

**Submitted to:**

**Name**: Md. Farmanul Haque

Assistant Professor

**Submitted by:**

**Name**: Suyash Srivastava

Roll Number: 191500835

**Name**: Shreya Yadav

Roll Number: 191500784

**INDEX**

1. Declaration
2. Abstract
3. Introduction
4. About The Project
5. Types of Recommendation Systems
6. Prediction
7. Technical Support
8. Introduction to Development Tools
9. Module Flowchart
10. User Interface
11. Certificate
12. Future Work
13. Conclusion
14. Reference

**DECLARATION**

We, the students of Computer Science and Engineering, GLA University, Mathura declare that the work entitled "MAVERICK: MOVIE RECOMMENDATION SYSTEM" has been successfully completed under the guidance of Prof. Md. Farmanul Haque, Computer Science and Engineering Department, GLA University, Mathura. This dissertation work is submitted in partial fulfilment of the requirements for the award of Degree of Bachelor of Engineering in Computer Science and Engineering during the academic year 2022-23.

Place: GLA University, Mathura

Date: 02/06/2022

Team members: Suyash Srivastava (191500835)

Shreya Yadav (191500784)

**ABSTRACT**

A movie recommendation is important in our social life due to its strength in providing enhanced entertainment. Our Project “Maverick: The Movie Recommendation System” can suggest a set of movies to users based on their interest, or the popularities of the movies.

Although, a set of movie recommendation systems have been proposed, most of these either cannot recommend a movie to the existing users efficiently or to a new user by any means. In this paper we propose a movie recommendation system that has the ability to recommend movies to a new user as well as the others.

It mines movie databases to collect all the important information, such as, popularity and attractiveness, required for recommendation. It generates movie swarms not only convenient for movie producer to plan a new movie but also useful for movie recommendation. Experimental studies on the real data reveal the efficiency and effectiveness of the proposed system.

**INTRODUCTION**

Given the huge number of movies are available all over the world, it is challenging for a user to find the appropriate movies suitable for his/her tastes. Different users like different movies or actors. It is important to find a method of filtering irrelevant movies and find a set of relevant movies.

Our Movie recommendation system project is a process of exactly doing above tasks. Such a system has lot of implications and is inspired by the success of recommendation systems in different domains such as books, TV program, jokes, news articles. It is one of the most important research projects in the digital television domain.

The most well-known recommendation systems are mainly based on Collaborative Filtering (CF) and Content-based Filtering. CF first tries to find out the groups of similar users automatically from a set of active users. The similarities between users are computed using correlation measure. It then recommends items to a user based on the opinions of the users’ groups. Although CF is successful in many domains, however, it has shortcomings such as, sparsity and scalability. CF uses user ratings to find similar users. However, it is very difficult to find such since very few movies have ratings.

**ABOUT THE PROJECT**

**Create**

: A Movie Recommendation System with the Help of Machine Learning & Python.

**Idea**

: The basic Idea for this project is to Use a Movie Dataset, Sort it in a manner that it provides the best possible 5 movie recommendations for the input movie.

**Goal**

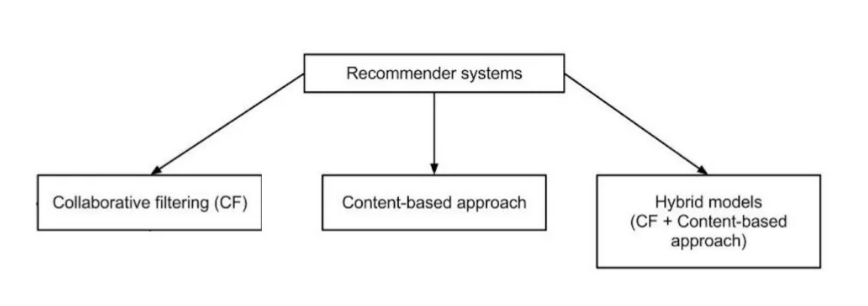
: The goal of the system is to provide personalized recommendations that help users find high quality videos relevant to their interests. In order to keep users entertained and engaged, it is imperative that these recommendations are updated regularly and reflect a user's recent activity on the site.

**Requirement**

: Recommender System is a system that seeks to predict or filter preferences according to the user's choices. Recommender systems are utilized in a variety of areas including movies, music, news, books, research articles, search queries, social tags, and products in general.

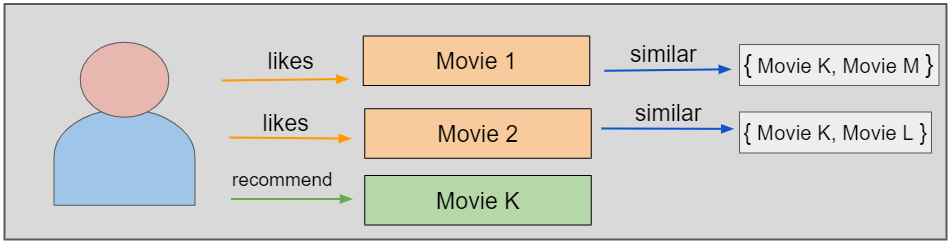
**Types of Recommender Systems**

Machine learning algorithms in recommender systems typically fit into two categories: content-based systems and collaborative filtering systems. Modern recommender systems combine both approaches.



**Content-Based Movie Recommendation Systems**

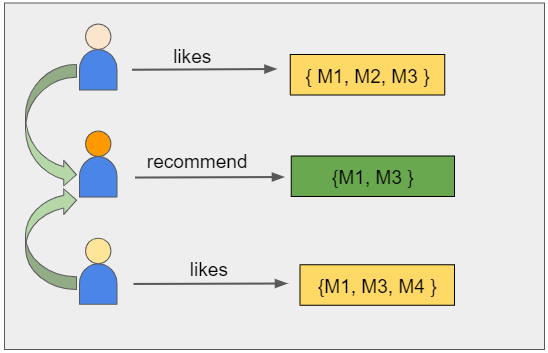
Content-based methods are based on the similarity of movie attributes. Using this type of recommender system, if a user watches one movie, similar movies are recommended. For example, if a user watches a comedy movie starring Adam Sandler, the system will recommend them movies in the same genre or starring the same actor, or both. With this in mind, the input for building a content-based recommender system is movie attributes.



# **Collaborative Filtering Movie Recommendation Systems**

With collaborative filtering, the system is based on past interactions between users and movies. With this in mind, the input for a collaborative filtering system is made up of past data of user interactions with the movies they watch.

For example, if user A watches M1, M2, and M3, and user B watches M1, M3, M4, we recommend M1 and M3 to a similar user C. You can see how this looks in the figure below for clearer reference.



**Prediction**

* Using regression we can solve for the weight vector, W
* User can input the movie for which he wants recommendation (say Oi)
* We check similarity, S(Oi, Oj) of the given movie with all other movies (Oj).
* Each movie's similarity score is dot\_product( S, W).
* We have to recommend movies which have the maximum similarity Score

**Technical Support**

1. **Technologies Used**

* Machine Learning
* Data Science

1. **Language Used**

* Python

1. **Development Platform**

* Jupyter Notebook
* PyCharm

1. **Deployment Platform**

* Heroku

1. **Server platform**

* Windows 7
* Windows 8
* Windows 10
* Windows 11

**Introduction to Development Tools**

1. **Jupyter Notebook**

The Jupyter Notebook is an open-source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at [Project Jupyter](http://jupyter.org/).

Jupyter Notebooks are a spin-off project from the Python project, which used to have a Python Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the Python kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.

**Benefits:**

* They’re great for showcasing your work. You can see both the code and the results. The notebooks at Kaggle is a particularly great example of this.
* It’s easy to use other people’s work as a starting point. You can run cell by cell to better get an understanding of what the code does.
* Very easy to host server side, which is useful for security purposes.

1. **PyCharm**

PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development.

**Features**

* Intelligent Coding Assistance. PyCharm provides smart code completion, code inspections, on-the-fly error highlighting and quick-fixes, along with automated code refactoring’s and rich navigation capabilities.
* Built-in Developer Tools.
* Web Development.
* Scientific Tools.
* Customizable and Cross-platform IDE.

1. **Kaggle**

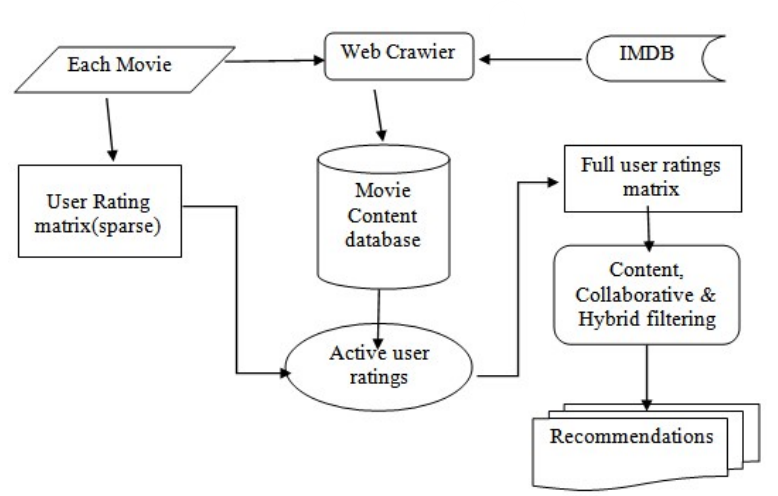
Kaggle allows users to find and publish data sets, explore and build models in a web-based data-science environment, work with other data scientists and machine learning engineers, and enter competitions to solve data science challenges.

**3.1 Dataset Used**

**TMDB 5000 Movie Dataset**

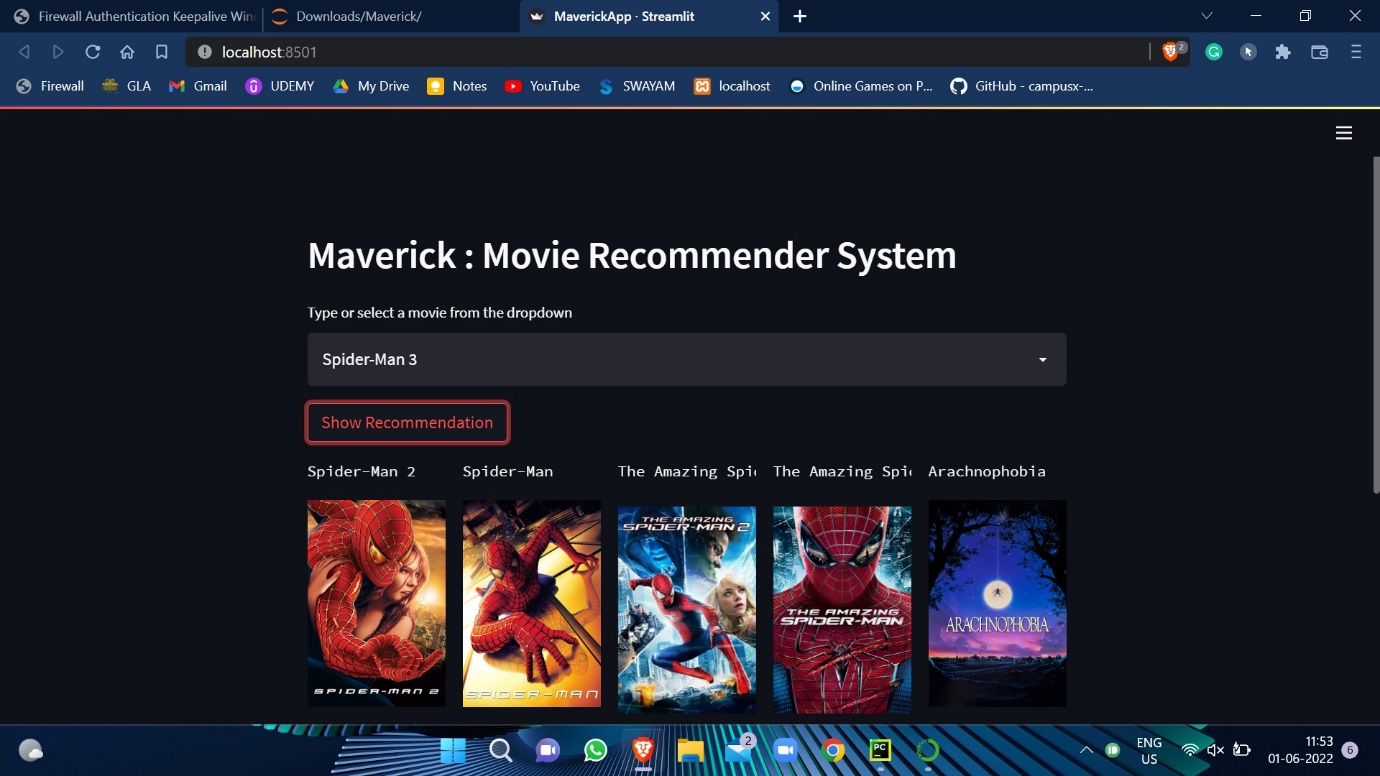
This dataset was generated from [The Movie Database](https://www.kaggle.com/datasets/tmdb/themoviedb.org) API. This product uses the TMDb API but is not endorsed or certified by TMDb.  
Their API also provides access to data on many additional movies, actors and actresses, crew members, and TV shows.

**Module Flowchart**



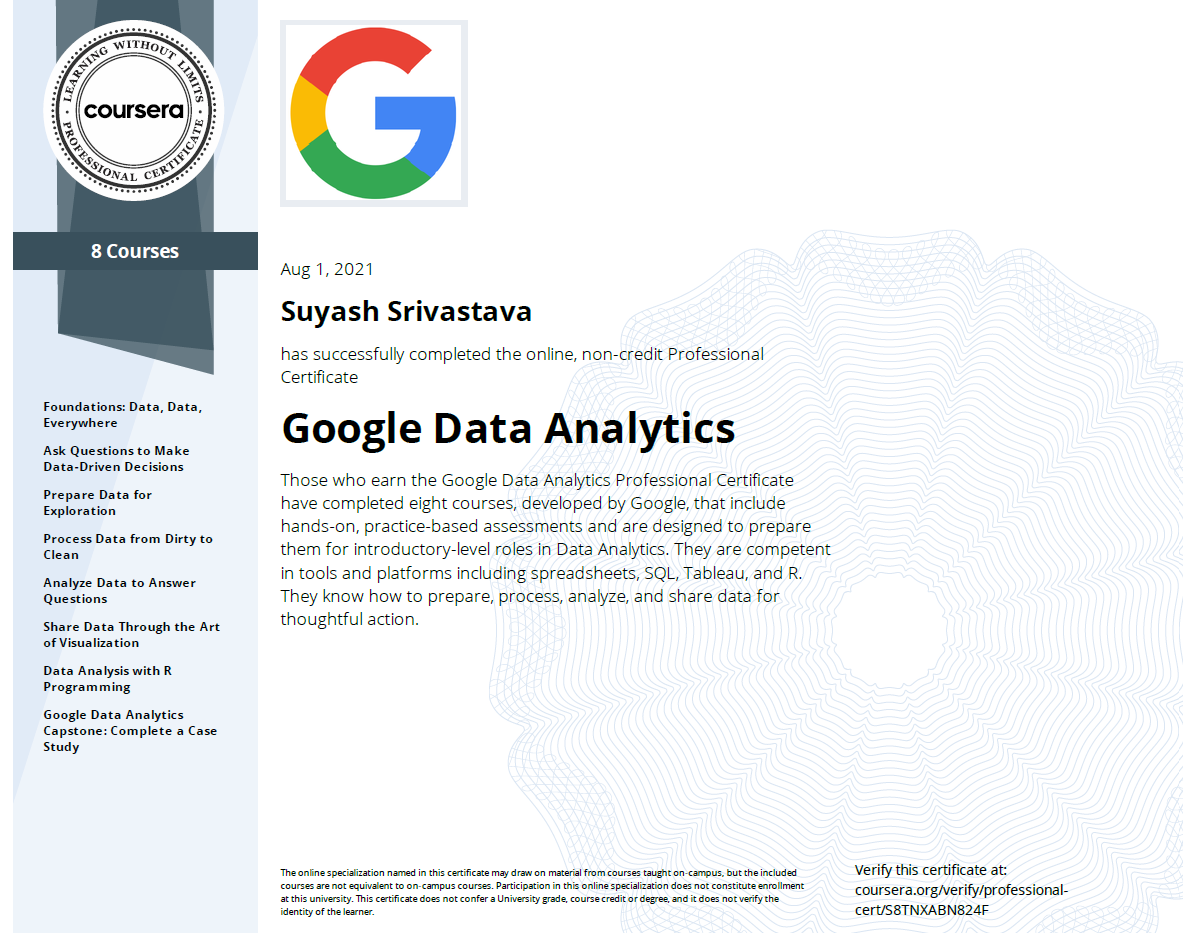
**User Interface**





**Certificates**





**Future Work**

* In collaborative filtering, we have a problem of sparsity of data. Very few users actually rate the same movie.
* We can use Clustering Algorithms like K-Means to cluster items or users or both based on their attributes.
* In the hybrid approach, we can use more features to get better predictions. (Currently, we have only 9 features)
* Deployment of Application upon Heroku.

**Conclusion**

In conclusion we believe this project if properly utilized will save time, reduce the amount of work the administration has to do. The system should also serve as a major tool to improving the efficiency in Movie Analysis. Hence a system with expected results has been developed but there is still room for improvement.

In terms of experience gained through the duration of this project study, the students have been able to have broader knowledge about the management of Movie Prediction using manual and automated procedures. The students have also been able to improve their knowledge in developing enterprise applications. We believe this project will serve the university efficiently in their efforts to automate the Movie Prediction Procedure.

**REFERENCE**

• [www.wikipedia.org/](http://www.wikipedia.org/•)

[•](http://www.wikipedia.org/•) [www.udemy.com/](http://www.udemy.com/)

• [www.google.com/](http://www.google.com/)

* www.coursera.com/