******

**PROJECT REPORT**

***on***

**MOVIE RECOMMENDER SYSTEM**

B. TECH in Artificial Intelligence and Data Science

2020-24



**Submitted to: Submitted by:**

Department of computer Applications **Saumya Deep Rawat** (16)

***University Roll Number:*** 2017666

*Under the guidance of* ***Dr Surender Singh Samant***

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

**GRAPHIC ERA DEEMED TO BE UNIVERSITY, DEHRADUN**

**DECLARATION**

I, Saumya Deep Rawat student of B-tech, Semester 5, Department of Computer Science and Engineering, Graphic Era Deemed to Be University, Dehradun, declare that the technical project work entitled “Movie Recommender System” has been carried out by me and submitted in partial fulfilment of the course requirements for the award of degree in B-tech of Graphic Era Deemed to Be University during the academic year 2022-23. The matter embodied in this synopsis has not been submitted to any other university or institution for the award of any other degree or diploma.

**ACKNOWLEDGEMENT**

I would like to take this opportunity to express my gratitude to entire faculty at Department of Computer Science and Information Technology, Graphic Era Deemed to Be University, Dehradun who evaluated the project from time to time and gave me the valuable suggestions as to how to improve the project.

I am grateful to ***Dr Surender Singh Samant***, Graphic Era Deemed to Be University, for his supervision, encouragement, inspiration, and guidance. Working under him is being an enriched experience. In all, I found congenial work environment in Graphic Era University, Dehradun and this project completion will mark a new beginning for me in the coming days.

I am highly indebted to Graphic Era University for providing me the required infrastructure and facilities to accomplish the given task.

Saumya Deep Rawat

Btech AI&DS

2020-24

Graphic Era University

**Motivation**

I was motivated to work on this project because I wanted to learn more about how to recommend movies to users. In particular, I wanted to learn about Content-Based Filtering and how to use it to make recommendations.

The main motivation for developing a movie recommender system is to provide users with personalized recommendations of movies that they might like. In recent years, there has been an increasing demand for such to the. recommender systems have the and effort in finding new movies to watch.

The tmdb5000 dataset is a great resource for building a movie recommender system. It contains a large amount of data that can be used to train the recommender system. Additionally, the dataset is well-organized and easy to use.

I believe that the tmdb5000 dataset is a great resource for building a movie recommender system because it is large and well-organized.

I was also motivated by the challenge of trying to recommend movies to users not rated many movies. I wanted to see if I could find a way to recommend movies to these users that they would enjoy.

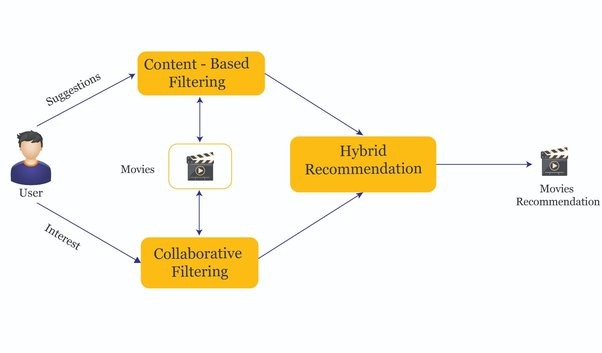
**Movie Recommender System**

**Problem Statement**

A movie recommender system takes in a user's movie preferences and outputs a list of recommended movies. The system should be able to recommend new movies to a user based on their past watch history. The system should also be able to handle new users and new movies. As such, the system should be constantly updated with the latest movies and user preferences. The system should also be able to handle a large number of users and a large number of movies. One issue is that recommender systems may suffer from the "cold start" problem, where they are unable to recommend anything to new users because they lack data on their preferences. To address these issues, some movie recommender systems use a hybrid approach, combining the collaborative filtering approach with content-based filtering. In content-based filtering, the system recommends movies based on their similarity to other movies that the user has liked. This approach overcomes the issue of the cold start problem, since the system can recommend similar movies even if it does not have any data on the user's preferences. However, content-based filtering has its own limitations, such as the need for a large amount of data in order to learn the user's preferences.

**Introduction**

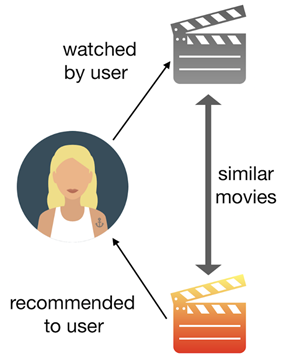
A recommender system is a type of information filtering system that seeks to predict the "rating" or "preference" a user would give to an item. Recommender systems are utilized in a variety of areas, with commonly recognized examples taking the form of playlist generators for video and music services, product recommenders for online stores, or content recommenders for social media platforms and open web content recommenders.

There are two types of movie recommender systems: content-based and collaborative-based. Content-based systems focus on the attributes of the movies, such as genre, director, and actors. Collaborative-based systems focus on the past behaviour of other users and make recommendations based on that. 

There are advantages and disadvantages to both types of systems. Content-based systems are better at making recommendations for movies that are similar to ones that the user has already seen. Collaborative-based systems are better at making recommendations for movies that the user might not have considered.

**Methodology**

For our project, we focused on Content-based filtering for generating recommendations.



Content-Based filtering doesn't involve other users, but is based on our preferences. It can be difficult to find new, interesting content. There are a few ways to get around this. One is to search for specific keywords that relate to your interests. This can help you find content that you might not have been able to find otherwise. If you're still having trouble finding new content, there are a few other options you can try.

we build this recommender system based on the following key features:

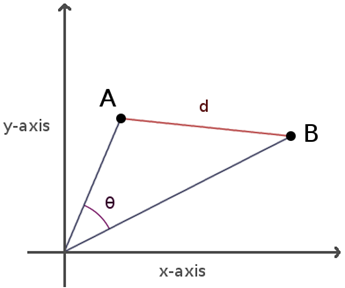
1 genre 4 cast

2 overview 5 crew

3 keywords

Now, moving forward to the term that we hold on mentioning, similarity, and what it approach in our context. It may not appear like some thing we could quantify, however it is able to be measured.

Cosine Similarity can be defined as a method of determining how similar two objects are. The cosine similarity is calculated using the cosine of the angle between two objects. The cosine similarity is also used in recommender systems and collaborative filtering. The cosine similarity is used to find the similarity between two vectors. The cosine similarity can also be used to measure the similarity between two users or items.



the area below the movie vectors A and B, represent the contents of the movies, and the angle, θ, between them represents the similarities between the movie contents. Thus, the lower the angle θ, the more similar the movie contents are.

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

**TOOLS USED**

**Google Collab:** Google collab is a software that allows users to collaborate on documents, spreadsheets, and presentations. It also allows users to chat with each other in real time.

**Pycharm:** Pycharm is a software used for programming in python. It has many features such as code completion and error checking. It also has a debugger which can be used to find and fix errors in your code. The community edition has all the basic features that you need for Python programming. If you are just starting out with Python, then the community edition is a good choice.

**Header File Used**

1. **Numpy**: Used for working with arrays.
2. **Pandas**: Used for data analysis.
3. **Sklearn**: Used for making use of Machine learning tools.
4. **AST:**This module helps python application to process trees of the python abstract syntax grammar.
5. **Pickle:** Pickle is used for converting Python objects into a byte stream, and vice versa. This is useful for sending data over a network, or for storing it in a file or database.

**Conclusion**

To conclude, a recommender system powered by content-based filtering performed using the cosine similarity algorithm can make better recommendations for users by suggesting them movies that have similar key features like the genre, keywords, overview, casts, directors etc.

**Project Repository**

Please visit my GitHub repository.

https://github.com/sdr999/Movie-recommender-system

**References**

https://youtu.be/1xtrIEwY\_zY