

## Abstract:

In the modern era of digital streaming platforms, providing personalized movie recommendations to users has become a critical aspect of enhancing user experience and engagement. This report presents the design, implementation, and evaluation of a machine learning-based movie recommendation system that leverages collaborative filtering techniques to generate personalized recommendations for users.

## Scope:

The movie recommendation system uses content-based filtering which is focused on movie attributes, such as genre, cast, and director, to generate recommendations and is limited to only suggesting movies from the dataset used to train the model. The application is built using the Flask framework in Python and is deployed locally on the user's system. It utilizes libraries such as Pandas and scikit-learn for data processing and ML algorithms. The movie recommendation engine takes user input and generates relevant movie suggestions by using a similarity matrix to find similar movies.

## Methodology:

1. Data Cleaning: First, we cleaned the dataset by removing any missing or duplicate values and fixed any inconsistencies in the data.
2. Exploratory Data Analysis: We performed EDA on the dataset to gain insights analyzing the distribution of various features (genres, release dates etc).
3. Feature Engineering: We applied text processing techniques (tokenization, stopword removal, and stemming) to convert textual data into numerical representations. Then, we created new features and transformed existing ones using CountVectorizer,
4. Machine Learning Models: We utilized processed textual data as input for content-based filtering algorithm (cosine similarity) to predict the success of recommended movies.

The GUI consists of: a) Homepage: which serves as the entry point for users and provides an overview of the application's features. b) CineMate: The choices page allows users to customize their movie recommendations by selecting three genres and five cast members upon which the page provides personalized suggestions. c) MovieMatchMaker: Here, Users can choose to receive recommendations based on their preferred genres, any specific release year or a specific movie title.

## Future Additions:

1. **Using Big Data techniques:** Using Apache Spark, Hadoop, and MongoDB, we can handle large-scale datasets and train more complex models, improving the quality of recommendations.
2. **Hybrid Recommendation Engine:** We could implement a hybrid recommendation engine that combines collaborative filtering with content-based filtering techniques allowing for more personalized recommendations to users.
3. **Natural Language Processing:** We can also use NLP techniques to enable users to search for movies using natural language queries improving the user experience. (techniques: named entity recognition or sentiment analysis)
4. **Context-Aware Recommendations:** Incorporating contextual information, such as, day of the year, or user location, to generate more relevant recommendations. For example, the system could recommend horror movies on Halloween or romantic comedies on Valentine's Day.
5. **Reinforcement Learning and Real-time data streaming using Kafka:** Reinforcement learning and real-time data streaming can enhance the recommendation system's personalization and accuracy by continuously adapting to user feedback. For example, if a user suddenly starts showing a preference for a new genre or actor, the system would quickly pick up on this and adjust the recommendations accordingly.

## Conclusion:

In conclusion, we developed a movie recommendation system that leverages content-based filtering techniques to generate personalized recommendations for users. We achieved this by performing data cleaning, exploratory data analysis, and feature engineering on the TMDB 5000 movie dataset. The system was implemented using the Flask framework in Python and utilized libraries like Pandas and scikit-learn for data processing and ML algorithms. The system's results showed that our model was able to generate accurate recommendations for users. The user feedback generated through peer checking was positive, indicating that the system provided a satisfactory user experience. Overall, the developed movie recommendation system can be a useful tool for digital streaming platforms to enhance user engagement and satisfaction.

## About Dataset: (TMDB 5000 movie dataset & TMDB 5000 credits dataset)

The dataset contains information about the movies' title, genre, budget, release date, and popularity, among other features.

*More details: <https://www.kaggle.com/datasets/tmdb/tmdb-movie-metadata>*