# **Data Science- Project Report**

# 

## **Movie Recommendation System**

### **Project Members**

### Roll no: 21k-3839

### Roll no: 21K-3863

### Roll no: 21K-3414

# **Movie Recommendation System - Project Report**

## **Introduction**

In today’s digital world, people seek personalized recommendations to quickly find relevant content among massive choices. A **Movie Recommendation System** serves this purpose by suggesting movies based on users' interests, preferences, and behavior. This project builds a simple yet effective movie recommendation system using data analysis and machine learning techniques. The application is developed using **Python**, **pandas**, **scikit-learn**, and **Streamlit** for web deployment.

The aim of this project is to create a system that recommends similar movies based on the movie the user selects, helping users discover new content they are likely to enjoy.

## **Objectives**

* Develop an efficient content-based movie recommendation engine.
* Process and clean movie metadata for accurate recommendations.
* Deploy the system with a simple and user-friendly interface.
* Handle large datasets effectively without performance degradation.

## **Dataset**

The project uses a public dataset consisting of metadata about movies, including:

**In tmdb\_5000\_movies.csv:**

Budget, genres, homepage, id, keywords, original\_language, original\_title, overview, popularity, production\_companies, production\_countries, release\_date, revenue, runtime, spoken\_languages, status, tagline, title, vote\_average, vote\_count

**In tmdb\_5000\_credits.csv:**

movie\_id, title, cast, crew

* Favorite movies genres
* Best director

## **Methodology**

**1. Data cleaning**:

* Handling missing values.
* Dropping or filling missing fields where necessary.
* Removing unnecessary columns like homepage, status, etc., that are not useful for recommendation

**2. Visualization:**

* Visualizing outliers
* Popular production companies
* Finding patterns in movies popularity over the years
* Top rated movies of the year

**3. Data Preprocessing**

* **Combining important fields** like genres, keywords, cast, and overview into a single feature named "tags".
* **Text cleaning**: Lowercasing and removing punctuation.
* **Stemming**: Applying stemming using NLTK's Porter Stemmer to normalize words.

**4. Vectorization**

* **Bag of Words (BoW)** model was used via CountVectorizer from scikit-learn to convert the textual tags into numerical vectors.
* Maximum 5000 frequent words were considered with the removal of English stopwords.

**5. Similarity Computation**

* **Cosine Similarity** was calculated between movie vectors to measure how similar two movies are.
* When a user selects a movie, the system finds the top 5–10 movies most similar to the selected movie.

## **System Architecture**

* **Backend**: Python with pandas, scikit-learn, nltk for data handling and machine learning tasks.
* **Frontend**: Streamlit was used to create an interactive web app.
* **Deployment**: The app is intended for deployment either on **Streamlit**

## **Features**

* **Searchable Dropdown**: Users can select a movie title from a list.
* **Recommended Movies**: Upon selection, the system displays a list of recommended movies.
* **Simple User Interface**: Built using Streamlit for fast and clean interaction.
* **Performance Optimized**: Pre-computed similarity matrix ensures real-time recommendations without delay.

## **Conclusion**

The **Movie Recommendation System** successfully demonstrates how basic natural language processing and machine learning techniques can be applied to develop an intelligent system. The project provides personalized movie suggestions based on content similarity, ensuring a better user experience in discovering movies. The scalable design of this system allows for further improvement and integration with larger real-world applications.

This project not only strengthens skills in data science, machine learning, and web deployment but also offers a strong foundation for building more advanced recommendation systems.