**Graphic Era Hill University**

**Dehradun**

**Mini Project Report on**

**“Movie Recommender System”**

**(CSE V Semester)**

**2024-2025**



**Submitted To: Submitted By:**

Dr. Vikrant Sharma Ishika Sharma

(CC-CSE-B2-V-SEM) Roll No. 2218878

CSE-B2-V-Sem

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**GRAPHIC ERA HILL UNIVERSITY, DEHRADUN**

**CERTIFICATE**

### Certified that Ishika Sharma (Roll No.- 2218878) have Completed Mini Project on “Movie Recommender System” based on Machine Learning for fulfilment of CSE V Semester Mini Project in Graphic Era Hill University, Dehradun. Student have successfully Completed this Course as best of my knowledge.

DATE: 11/01/2024

(Dr. Vikrant Sharma)

**Class Coordinator**

**CC-CSE-B2-V-Sem**

CSE Department

GEHU, DEHRADUN

**ACKNOWLEDGEMENT**

I wish to thank my parents for their continuing support and encouragement. We also wish to thank them for providing us with the opportunity to reach this far in our studies.

I would like to thank my classmates and friends for their patience, support and encouragement throughout the completion of this course.

I also acknowledge to my Subject Teacher and Class coordinator Dr. Vikrant Sharma Sir who helped me to complete this project.

At last but not the least I greatly indebted to all other persons who directly or indirectly helped me during this project.

**Ishika Sharma**

**Roll No.- 2218878**

**CSE-B2-V-Sem**

**Session: 2024-2025**

**GEHU, Dehradun**

**TABLE OF CONTENTS**

1. Introduction to Movie Recommender Systems  
    1.1 Definition  
    1.2 Key Features
2. Recommender Systems Overview  
    2.1 Definition  
    2.2 Types of Recommender Systems
3. Machine Learning Overview  
    3.1 Definition  
    3.2 Types of Machine Learning  
    3.3 Application to Movie Recommender Systems
4. Data Collection & Preparation  
    4.1 Data Sources  
    4.2 Data Cleaning  
    4.3 Feature Engineering  
    4.4 Exploratory Data Analysis (EDA)
5. Model Building  
    5.1 Algorithm Selection  
    5.2 Evaluation Metrics  
    5.3 Optimization
6. Training & Evaluation  
    6.1 Training  
    6.2 Validation  
    6.3 Cross-Validation  
    6.4 Evaluation Metrics
7. User Interface with Streamlit  
    7.1 Interactive Dashboard  
    7.2 Features  
    7.3 Visualization
8. Deployment  
    8.1 Google Colab  
    8.2 Streamlit Sharing  
    8.3 Scalability
9. Conclusion

**1. INTRODUCTION TO MOVIE RECOMMENDER SYSTEMS**

1.1 **Definition**

A movie recommender system is a machine learning-based application that predicts and suggests movies to users based on their preferences and viewing history. It enhances user experience by providing personalized recommendations.

1.2 **Key Features**

* **Personalization:** Tailors suggestions based on individual user preferences.
* **Scalability:** Handles large datasets efficiently to cater to multiple users.
* **Real-Time Recommendations:** Provides instant and dynamic suggestions based on user interactions.

**2. RECOMMENDER SYSTEMS OVERVIEW**

2.1 **Definition**

Recommender systems are algorithms designed to suggest relevant items to users, such as movies, books, or products. They analyze user behavior and patterns to predict preferences.

2.2 **Types of Recommender Systems**

* **Content-Based Filtering:** Recommends items similar to those the user has liked previously.
* **Collaborative Filtering:** Suggests items based on the preferences of similar users.
* **Hybrid Systems:** Combines multiple approaches to improve accuracy.

**3. MACHINE LEARNING OVERVIEW**

3.1 **Definition**

Machine Learning (ML) is a subset of artificial intelligence that enables systems to learn and improve from experience without explicit programming. It involves algorithms that identify patterns in data to make predictions or decisions.

3.2 **Types of Machine Learning**

* **Supervised Learning:** Models are trained using labeled data to predict outcomes (e.g., predicting movie ratings).
* **Unsupervised Learning:** Identifies patterns or groupings in unlabeled data (e.g., clustering similar users based on preferences).
* **Reinforcement Learning:** Models learn through trial and error to maximize rewards (e.g., optimizing recommendations based on user feedback).

3.3 **Application to Movie Recommender Systems**

This project primarily utilizes supervised and unsupervised learning techniques:

* **Supervised Learning:** To predict user ratings or preferences for specific movies.
* **Unsupervised Learning:** To cluster users or movies into groups for better recommendations.

**4. DATA COLLECTION & PREPARATION**

4.1 **Data Sources**

Use publicly available datasets like MovieLens or IMDb.

4.2 **Data Cleaning**

Handle missing values, duplicates, and outliers.

4.3 **Feature Engineering**

Create meaningful features like genre preferences, ratings, and watch history.

4.4 **Exploratory Data Analysis (EDA)**

Visualize and understand the data distribution.

**5. MODEL BUILDING**

5.1 **Algorithm Selection**

Implement collaborative filtering and content-based filtering using Python libraries like scikit-learn.

5.2 **Evaluation Metrics**

Use precision, recall, and F1-score to measure model performance.

5.3 **Optimization**

Fine-tune hyperparameters to improve prediction accuracy.

**6. TRAINING & EVALUATION**

6.1 **Training**

Use training data to build models and adjust weights based on loss functions.

6.2 **Validation**

Split data into training and validation sets to monitor overfitting.

6.3 **Cross-Validation**

Employ k-fold cross-validation for robust performance evaluation.

6.4 **Evaluation Metrics**

Analyze performance using RMSE (Root Mean Square Error) and MAE (Mean Absolute Error).

**7. USER INTERFACE WITH STREAMlit**

7.1 **Interactive Dashboard**

Design a user-friendly interface using Streamlit.

7.2 **Features**

Allow users to input preferences, view recommendations, and filter results by genre or release year.

7.3 **Visualization**

Display user activity trends and recommendation accuracy.

**8. DEPLOYMENT**

8.1 **Google Colab**

Use Google Colab for model training and testing.

8.2 **Streamlit Sharing**

Deploy the application using Streamlit’s cloud platform.

8.3 **Scalability**

Plan for future integration with larger datasets and real-world platforms.

**9. CONCLUSION**

Developing a movie recommender system involves leveraging machine learning techniques to enhance user experience. By combining effective planning, robust model development, and an interactive Streamlit interface, the system can provide valuable, personalized recommendations. Leveraging resources like Google Colab and open datasets ensures accessibility and scalability. Future improvements could include incorporating deep learning techniques and real-time user feedback for enhanced accuracy.