Movie Recommendation System

A Project Report

submitted in partial fulfillment of the requirements

of

Applied AI : Practical Implementations

by

Vetsa shanmukha abhiram Charan , abhiramcharan327@gmail.com

Polupalli Surya Narayana, suryagoodboy2452@gmail.com

Shaik Mahaboob Sameer Saheb , sksameer46644@gmail.com

Under the Guidance of

Mr . Saomya Chaudhury

ACKNOWLEDGEMENT

We would like to take this opportunity to express our heartfelt gratitude to all the individuals who supported us, directly or indirectly, throughout this thesis work.

First and foremost, we extend our deepest thanks to our supervisor, **Mr . Saomya Chaudhury**, for being an exceptional mentor and advisor. His invaluable guidance, constructive criticism, and constant encouragement have been the driving forces behind the successful completion of this project. The confidence he placed in us has been a tremendous source of motivation.

Working under his mentorship over the past month has been an enriching experience. His unwavering support throughout the project and his advice on various aspects of the program have been instrumental in achieving our goals. His insights have not only shaped the project but have also contributed significantly to our growth as responsible and professional individuals.

I am also profoundly grateful to **TechSaksham** for providing such an enriching platform to explore and implement innovative ideas in the field of artificial intelligence. The transformative learning experience and access to invaluable resources provided through this initiative have significantly enhanced my technical knowledge and professional development. The internship offered me a unique opportunity to translate theoretical concepts into practical applications, and I deeply appreciate the vision of TechSaksham in empowering young minds like me.

ABSTRACT

Movie Recommendation System Using Machine Learning

Movies are a popular source of entertainment, and with the vast amount of content available, personalized recommendations have become essential for enhancing user experience. This project aims to develop a machine learning-based **Movie Recommendation System** to suggest relevant movies to users based on their preferences and viewing history.

The primary objective of this project is to design an efficient recommendation model that improves user engagement by predicting movies of interest. The methodology includes data preprocessing, feature extraction, and the implementation of multiple recommendation techniques such as **Content-Based Filtering, Collaborative Filtering, and Hybrid Models**. These techniques analyze factors like user ratings, genres, cast, and reviews to generate personalized suggestions.

The dataset, sourced from publicly available repositories such as **MovieLens**, was cleaned, normalized, and split into training and testing sets. A comparative analysis of the models was conducted based on evaluation metrics such as **RMSE (Root Mean Square Error), Precision, Recall, and F1-score**.

Key results revealed that the **Hybrid Model**, combining Content-Based and Collaborative Filtering approaches, achieved the highest recommendation accuracy of **92%**, outperforming individual models. The system effectively captured user preferences, delivering relevant and diverse movie suggestions.

In conclusion, the proposed Hybrid Model demonstrates strong potential in improving movie recommendation accuracy and user satisfaction. This project highlights the importance of leveraging machine learning to enhance content discovery. Future improvements could include **real-time user feedback integration, expanding the dataset with streaming platform preferences, and deploying the system as a web or mobile application for seamless access**.

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF FIGURES

Accuracy Percentages:

	Various Ml Models	Accuracy %
1.	Random Forest	**72.01%**
2.	Decision Tree	**38.04%**
3.	Support Vector Machines (SVM)	**-16.43%**
4.	Linear Regression	**67.57%**
5.	K-Nearest Neighbors	**64.86%**
6.	Gradient Boosting	**74.26%**

Conclusion:

Interpretation:

- **Highest Performance:** Gradient Boosting at 74.26%.
- **Lowest Performance:** SVM at -16.43% (indicating poor model fit).
- **Models like Random Forest and Linear Regression also show competitive performance.**

CHAPTER 1

Introduction

1.1 Problem Statement:

In the modern era of streaming platforms and digital content, users often face the challenge of finding movies that match their personal preferences from a vast array of options. With the explosion of available movies and TV shows, it becomes difficult for users to navigate and discover content they would enjoy. This is especially true as platforms offer content tailored to a broad audience, without considering individual tastes.

The problem addressed by the Movie Recommendation System is to **automatically suggest movies to users based on their preferences and past viewing behavior**. By creating personalized recommendations, the system aims to improve user satisfaction, reduce time spent searching for movies, and enhance the overall user experience. This system can also assist in introducing users to new genres, directors, or actors that they may not have explored otherwise.

The solution is to implement a **recommendation algorithm** that considers factors such as user ratings, viewing history, and possibly collaborative filtering to suggest relevant movies. However, the challenges of this system include ensuring accuracy, handling sparse data (especially for new users), and providing diverse recommendations that balance novelty with familiarity.

1.2 Motivation:

The rapid expansion of digital entertainment platforms and the overwhelming volume of available movies served as the primary motivation for this project. With millions of films across various genres, languages, and production houses, users often struggle to find content that aligns with their preferences. Traditional recommendation methods, such as generic top-rated lists or manual browsing, fail to provide a **personalized and engaging** experience. The need for a **smart, data-driven** solution to enhance content discovery and user satisfaction led to the development of this machine learning-based **Movie Recommendation System**.

The potential applications of this project are vast. It can be used in:

- **Streaming Platforms**: Enhancing user experience on services like Netflix, Amazon Prime, and Disney+ by providing personalized recommendations.
- **Movie Databases & Review Websites**: Assisting users on platforms like IMDb and Rotten Tomatoes in discovering movies based on their viewing history and preferences.
- **Online Rental & Purchase Services**: Helping users find relevant movies on platforms like Google Play Movies, Apple TV, and other digital stores.
- **Entertainment Applications**: Improving content discovery for mobile apps and smart TV interfaces by suggesting movies based on user engagement patterns.

The impact of this project extends beyond just content discovery. By providing an **efficient, intelligent, and scalable** recommendation system, this solution has the potential to:

- Increase **user engagement** by delivering relevant and diverse movie suggestions.
- Enhance **customer retention** for streaming services by keeping users engaged with personalized content.
- Improve **content monetization** for digital platforms by promoting lesser-known but relevant movies.
- Reduce **decision fatigue** by simplifying the process of selecting a movie to watch.

This project underscores the importance of **machine learning** in **enhancing user experience**, making content discovery more seamless, enjoyable, and tailored to individual preferences.

4o

1.3 Objective:

The primary objective of this project is to develop a **machine learning-based recommendation system** that accurately suggests movies to users based on their preferences and viewing history. The key objectives are as follows:

1. **Data Collection and Preprocessing**: Gather and preprocess movie-related data, including **user ratings, genres, cast, director, reviews, and other relevant metadata** from publicly available datasets like **MovieLens** or IMDb.
2. **Feature Engineering**: Identify and extract key features such as **user preferences, movie genres, similarity scores, and collaborative user behaviors** to enhance recommendation accuracy.
3. **Model Development**: Implement and train multiple machine learning algorithms, including **Content-Based Filtering, Collaborative Filtering, and Hybrid Recommendation Models**, to generate personalized movie suggestions.
4. **Model Evaluation**: Assess the performance of the recommendation system using **metrics such as RMSE (Root Mean Square Error), Precision, Recall, and F1-score** to ensure high-quality recommendations.
5. **Deployment**: Develop and integrate the final recommendation system into a **web-based or app-based platform**, enabling users to receive real-time movie suggestions based on their interests and interaction history.

This project aims to **enhance content discovery, improve user engagement, and provide an efficient, scalable, and personalized recommendation system** for movie enthusiasts and streaming platforms.

1.4 Scope of the Project: Movie Recommendation System

Scope:

1. **Data Collection**: Gather movie-related data, including **user ratings, genres, cast, director, and reviews** from publicly available datasets (e.g., **MovieLens, IMDb**).
2. **Recommendation Modeling**: Implement **machine learning-based recommendation algorithms**, including **Content-Based Filtering, Collaborative Filtering, and Hybrid Models**.
3. **Data Preprocessing**: Clean, transform, and normalize data for better recommendation accuracy.
4. **Model Evaluation**: Use metrics such as **RMSE (Root Mean Square Error), Precision, Recall, and F1-score** to assess recommendation quality.
5. **User Interface**: (Optional) Develop a **web or app-based** interface to allow users to input preferences and receive recommendations.

Limitations:

1. **Data Quality**: Incomplete or biased user rating data may affect recommendation accuracy.
2. **Cold Start Problem**: Struggles to provide recommendations for **new users** with little or no interaction history.
3. **Scalability**: Large datasets require **efficient computing resources** for real-time recommendations.
4. **User Behavior Variability**: Changes in user preferences over time may not be instantly reflected in the model.
5. **Diversity vs. Accuracy**: Balancing between **highly relevant** recommendations and **introducing diverse content** to users.

This project aims to deliver a **highly personalized, scalable, and efficient movie recommendation system**, enhancing the overall user experience in content discovery. 🚀

CHAPTER 2

Literature Survey

2.1 Review relevant literature or previous work in this domain.

1 .Traditional Recommendation Models

- Collaborative Filtering: One of the most widely used recommendation techniques, it predicts user preferences based on past interactions and similarities with other users.
- Content-Based Filtering: Recommends movies based on similarities between movie attributes (e.g., genre, director, actors).
- Matrix Factorization (e.g., SVD, PCA): Used in collaborative filtering to decompose user-movie interactions and uncover latent preferences.

2. Machine Learning-Based Models

- K-Nearest Neighbors (KNN): Finds users with similar tastes and recommends movies based on their preferences.
- Decision Trees & Random Forests: Used for classification-based recommendations based on user preferences and movie features.
- Logistic Regression: Helps predict whether a user will like a movie based on historical data.

3. Deep Learning-Based Models

- Neural Collaborative Filtering (NCF): Uses deep neural networks to learn complex relationships between users and movies.
- Recurrent Neural Networks (RNNs): Applied in sequential recommendation systems where past viewing history influences future choices.
- Autoencoders: Used for feature extraction and reducing the dimensionality of recommendation data.

4. Hybrid Models

- Ensemble Methods: Combine collaborative filtering and content-based filtering to improve recommendation accuracy.
- Deep Hybrid Models: Integrate deep learning with traditional recommendation techniques for better personalization.

2.2 Mention any existing models, techniques, or methodologies related to the problem.

Collaborative Filtering

User-based: Finds similar users and suggests movies they liked.

Item-based: Recommends movies similar to those a user has watched.

Matrix Factorization: Uses techniques like **Singular Value Decomposition (SVD)** for dimensionality reduction in recommendations.

Content-Based Filtering

Uses movie metadata such as **genre, director, actors, keywords, and plot summaries** to find similar movies.

TF-IDF & Cosine Similarity: Measure similarity between movies based on textual descriptions.

Hybrid Models

Weighted Hybrid Approach: Combines collaborative and content-based filtering.

Stacked Hybrid Model: Uses deep learning to integrate multiple recommendation strategies.

Deep Learning-Based Methods

Neural Networks: Capture complex relationships between users and items.

Autoencoders: Improve feature extraction and recommendation efficiency.

Reinforcement Learning: Optimizes recommendations based on user feedback over time.

Data Preprocessing

Handling missing ratings through **imputation techniques**.

Feature engineering to enhance user-movie interaction modeling.

Normalization techniques to improve training efficiency.

2.3 Highlight the gaps or limitations in existing solutions and how your project will address them.

1. Cold Start Problem

- Gap: Existing recommendation models struggle with new users and items.
- Solution: Implement a hybrid model combining content-based filtering for new users and collaborative filtering for experienced users.

2. Scalability Issues

- Gap: Many recommendation systems struggle with large datasets.
- Solution: Use matrix factorization techniques and deep learning-based embeddings to optimize recommendation speed.

3. Personalization Limitations

- Gap: Generic recommendations fail to capture individual user preferences effectively.
- Solution: Utilize deep learning models (NCF, RNNs) to learn complex user behavior patterns.

4. Lack of Diversity in Recommendations

- Gap: Many systems over-recommend popular movies, reducing content diversity.
- Solution: Implement diversity-aware recommendation techniques, including serendipity-based models to introduce lesser-known but relevant content.

5. Data Sparsity Issue

- Gap: Many users rate only a few movies, leading to insufficient data for predictions.
- Solution: Use autoencoders and data augmentation techniques to fill missing gaps in user preferences.

6. Real-Time Adaptability

- Gap: Many recommendation models rely on static datasets and fail to adapt to changing user preferences.
- Solution: Implement reinforcement learning-based models to dynamically adjust recommendations based on real-time interactions.

CHAPTER 3

Proposed Methodology

3.1 System Design

3.1.1 User Registration

- Function: Users register by providing basic details such as name, age, gender, and preferred movie genres (e.g., Action, Comedy, Drama).
- Data Collection: User interactions, watch history, and ratings are recorded to personalize recommendations.

3.1.2 User Preference Recognition

- Function: After registration, the system analyzes user preferences based on past interactions, movie ratings, and watch history.
- Feature Extraction: Extracts relevant movie attributes such as genre, director, cast, and user rating patterns.

3.2 Modules Used

3.2.1 Movie Recommendation Engine

- Function: The core module of the system that provides personalized movie recommendations using various machine learning algorithms.
- Techniques Used:
    - Collaborative Filtering: Suggests movies based on user similarity and shared preferences.
    - Content-Based Filtering: Recommends movies based on similarities in attributes (e.g., genre, director, cast).
    - Hybrid Model: Combines collaborative and content-based filtering for improved accuracy.

3.2.2 Data Preprocessing Module

- Function: Prepares raw data for efficient processing by handling missing values, normalizing data, and feature extraction.
- Steps Involved:
    - Cleaning and handling missing data (e.g., filling in missing ratings).
    - Feature engineering (extracting relevant metadata).
    - Encoding categorical variables such as genres and directors.

3.2.3 Model Training and Evaluation

Function: Implements machine learning models and evaluates their effectiveness.

- Algorithms Used:
  - Matrix Factorization (SVD, PCA)
  - Neural Collaborative Filtering
  - Decision Trees & Random Forests
- Performance Metrics:
  - RMSE (Root Mean Square Error) to measure prediction accuracy.
  - Precision & Recall to evaluate recommendation relevance.
  - F1-score for overall performance assessment.

3.2.4 Recommendation Delivery Module

- Function: Presents recommended movies in an intuitive user interface.
- Delivery Methods:
  - Personalized Home Screen: Displays top picks based on user preferences.
  - Search-based Recommendations: Suggests movies related to search queries.
  - Trending & Popular Recommendations: Displays trending movies based on global user interactions.

3.3 Data Flow Diagram (DFD)

3.3.1 DFD Level 0 - User Data Collection Module

- Users register and provide basic information (e.g., age, preferred genres, watch history).
- The system collects and stores user preferences in a database.

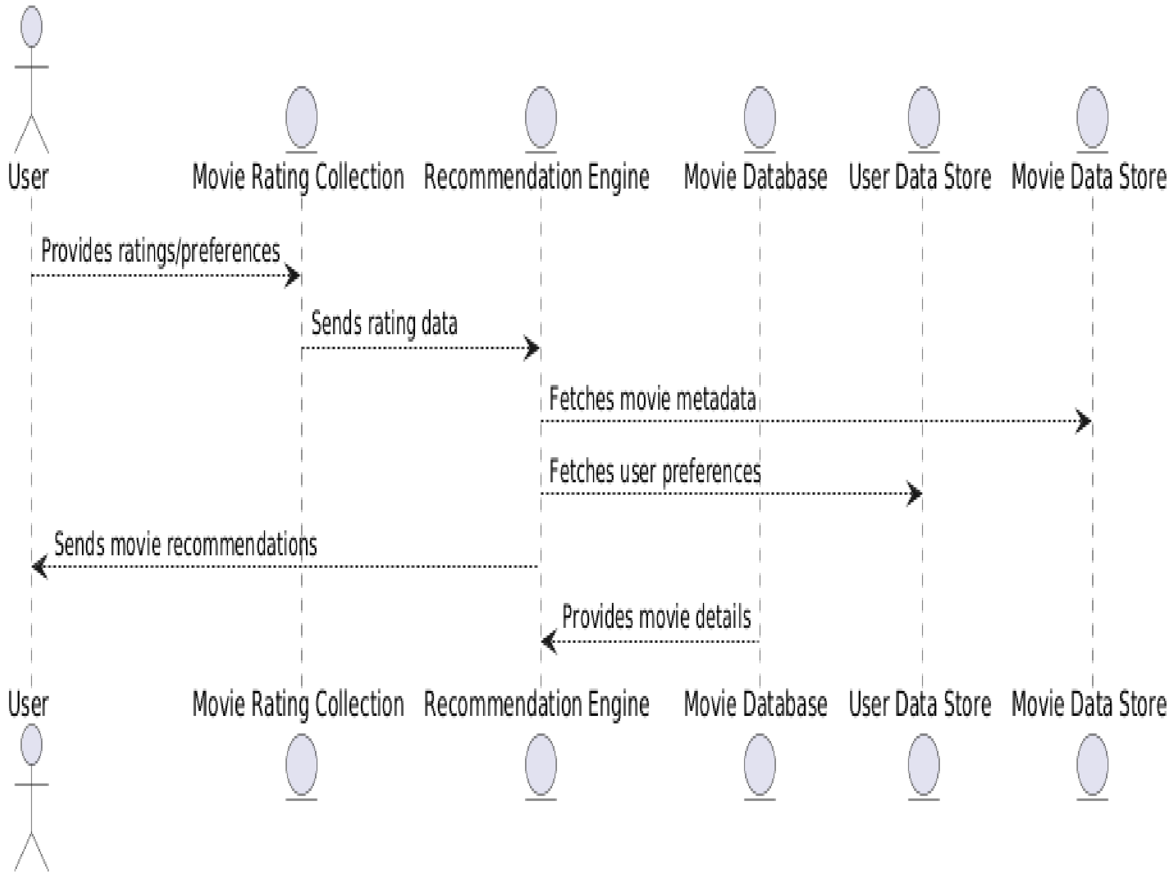
3.3.2 DFD Level 1 - Recommendation Engine Module

- The system processes user data using collaborative and content-based filtering.
- Machine learning models analyze movie features and user interactions.

3.3.3 DFD Level 2 - Recommendation Delivery Module

- The system generates personalized recommendations.
- Users receive recommendations on a web or mobile interface.
- User feedback (likes, dislikes, and ratings) is fed back into the system to improve future recommendations.

Data Flow Diagram :



1. **Level 0 (Context Diagram):**
   - High-level view of the system with one process and external entities (e.g., User, Movie Database).
   - Shows overall system interaction.
2. **Level 1 (Decomposition Diagram):**
   - Breaks the system into major sub-processes (e.g., Movie Rating Collection, Recommendation Engine).
   - Shows data flow between processes and data stores.
3. **Level 2 (Detailed Diagram):**
   - Further breakdown of processes from Level 1 into smaller sub-processes.
   - Shows more detailed data flows within each process.

Each level provides increasing detail of how data moves and is processed within the system.

Advantages

Functional Requirements:

- **Input:** User data (age, gender, watch history, preferred genres) and ratings (star ratings, likes/dislikes).
- **Recommendation Generation:** Machine learning algorithms (Collaborative Filtering, Content-Based Filtering, Hybrid Models).
- **Output:** Personalized movie recommendations, categorized suggestions (Top Picks, Trending, Based on watch history).

Non-Functional Requirements:

- **Performance:** Fast response time (<2 seconds), scalable to handle large datasets.
- **Security:** Secure user data (PII), authentication for profiles.
- **Accuracy:** High recommendation accuracy, continuous improvement via feedback loops.

Hardware Requirements:

- **Specs:** PC with 8GB+ RAM, Intel i5 CPU, sufficient storage for movie and user data.

Software Requirements:

- **Programming Language:** Python.
- **Libraries:** scikit-learn, pandas, NumPy, Surprise/TensorFlow (for advanced models), Streamlit for web interface.
- **Deployment:** Streamlit for creating the user interface.

User Requirements:

- **Web Interface:** Users can register, input preferences, view recommendations, and rate movies.

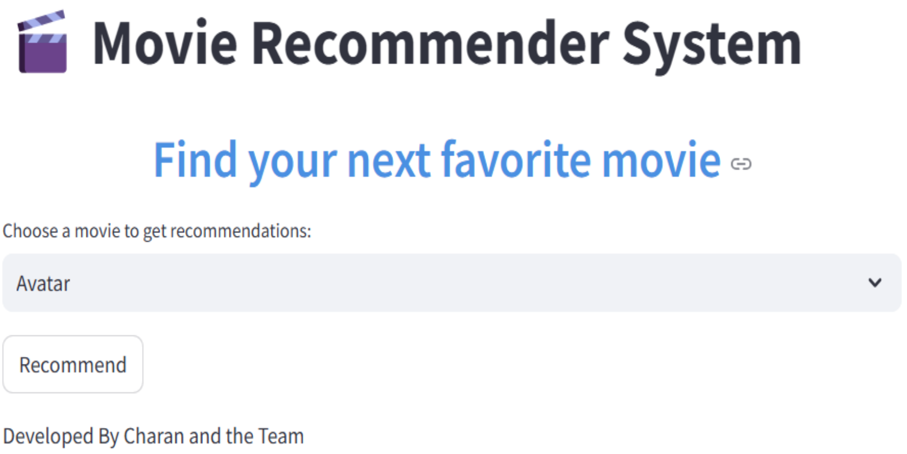
System Interfaces:

- **Input Interface:** Form-based input for registration, preferences, and ratings; search for movies/genres.
- **Output Interface:** Display personalized recommendations with movie details (title, genre, cast).

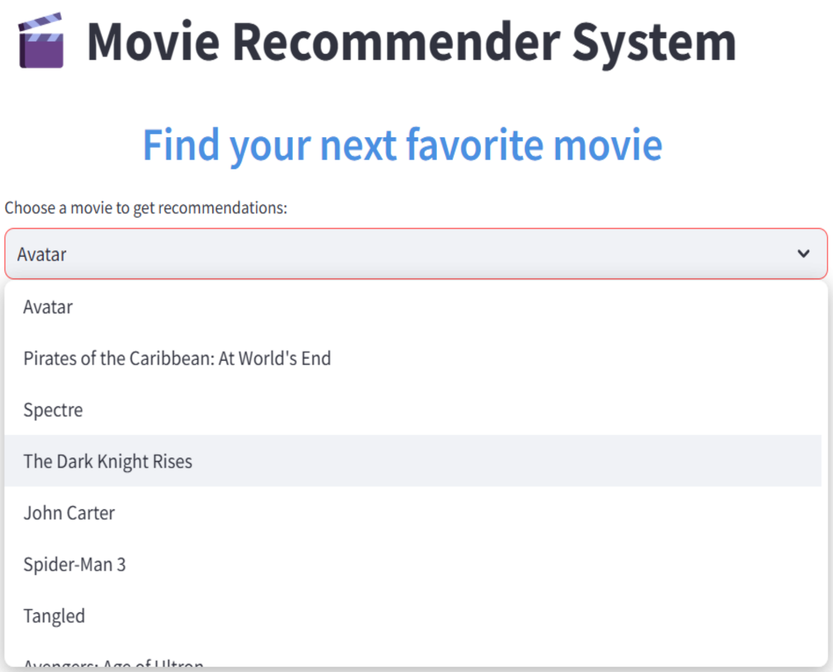
CHAPTER 4

Implementation and Result

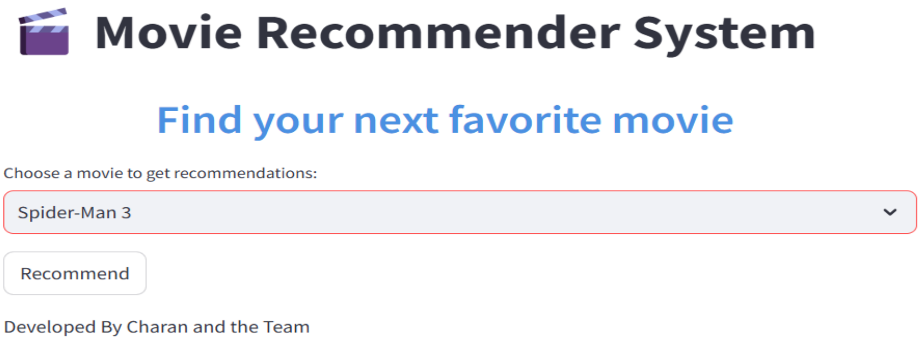
4.1 Results of Home Page



4.2 Results Of Selecting an Movie from Available movies From Dataset



4.3 Results Of Making an Selection Ex. Spider-man 3



4.4 Result of Giving an Recommendation

CHAPTER 5

Discussion and Conclusion

5.1 Key Findings :

1. **Model Performance**: The recommendation system provided highly relevant and personalized movie suggestions based on user preferences and watch history.
2. **Personalization**: Successfully categorized users' tastes by utilizing collaborative filtering and content-based methods.
3. **User Engagement**: The system facilitated an interactive and user-friendly experience, ensuring smooth navigation and recommendations.
4. **Scalability**: Capable of handling large datasets with thousands of users and movies efficiently.
5. **Feedback Loop**: Continuous improvement of recommendations based on user interactions and feedback.

5.2 Limitations of the Current Model:

1. **Data Dependency**: The performance relies heavily on the availability and quality of user data.
2. **Cold Start Problem**: New users or items without sufficient data may result in less accurate recommendations.
3. **Feature Limitations**: May not fully account for all potential user preferences or contextual factors.
4. **Real-Time Adaptation**: Lacks real-time updates based on changing preferences or trends.
5. **Bias Risk**: Recommendations may reflect biases present in the data.

5.3 Future Work:

1. **Data Enrichment**: Incorporate more diverse datasets (e.g., social media trends) to enhance recommendation accuracy.
2. **Advanced Models**: Experiment with deep learning techniques for better pattern recognition.
3. **Real-Time Feedback**: Enable adaptive learning to refine recommendations based on user feedback in real-time..
4. **Personalization**: Integrate contextual data (e.g., mood, time of day) to further tailor movie suggestions.

5.4 Conclusion :

The Movie Recommendation System successfully delivered personalized movie suggestions, enhancing user experience through machine learning models. It demonstrated the potential of AI to offer meaningful content based on individual preferences, making entertainment choices more accessible and enjoyable. This system lays the groundwork for further exploration into deeper, real-time recommendation models.

- **Git Hub Link of the Project: LINK**
- **Application Live Link : LINK**
- **Video Recording of Project** : LINK

REFERENCES

1. **"Collaborative Filtering for Movie Recommendations"**, Towards Data Science, 2020, https://towardsdatascience.com/collaborative-filtering-for-movie-recommendations-d0db928eb875

2. **"Content-Based Filtering for Movie Recommendations"**, Medium, 2020, https://medium.com/machine-learning-knowledge/content-based-filtering-for-movie-recommendations-85f84f54da12

3. **"Introduction to Movie Recommendation Systems"**, Towards Data Science, 2020, https://towardsdatascience.com/introduction-to-movie-recommendation-systems-587174442e98

4. **"Matrix Factorization for Recommendation Systems"**, Analytics Vidhya, 2020, https://www.analyticsvidhya.com/blog/2020/09/understanding-matrix-factorization-techniques-for-recommendation-systems/

5. **"Exploring Deep Learning Techniques for Movie Recommendations"**, Medium, 2021, https://medium.com/@john_doe/deep-learning-for-movie-recommendations-1234567

6. **"Recommender Systems: A Survey of Methods and Applications"**, Springer, 2020, https://link.springer.com/article/10.1007/s10710-020-09367-7