Artificial Intelligence and Machine Learning

Project Report

Semester-IV (Batch-2022)

MOVIE RECOMMENDATION SYSTEM

Submitted To:

Ms. Monica Dutta

Submitted By:

Mannat (2210990557)

Manya (2210990561)

Manya (2210990562)

Chitkara University Institute of Engineering & Technology,
Chitkara University, Punjab

ABSTRACT

This Python project implements a movie recommender system utilizing collaborative filtering and content-based filtering techniques. The system analyzes user preferences and movie attributes to generate personalized recommendations for users. It addresses challenges such as the cold start problem and sparsity of data through innovative algorithms and data preprocessing methods. The project aims to provide an efficient and user-friendly tool for movie enthusiasts to discover new and relevant content tailored to their tastes.

The system utilizes collaborative filtering techniques and methods to analyse user preferences and recommend relevant movies. Collaborative filtering leverages user-item interactions to identify similar users and recommend movies liked by those with similar tastes.

Overall, this movie recommender system represents a significant advancement in recommendation technology, offering a seamless and enjoyable movie discovery experience for users while empowering content providers with enhanced user engagement and retention strategies.

TABLE OF CONTENTS

INTRODUCTION:

1.1 BACKGROUND

The process of movie selection and recommendation is fundamental to the entertainment industry, influencing both viewers and content providers alike. Traditionally, movie recommendations have relied on manual curation processes, often based on subjective opinions and historical viewing patterns. However, these traditional methods have limitations, including the inability to analyse vast amounts of data and identify nuanced viewer preferences.

In recent years, there has been a growing interest in leveraging artificial intelligence (AI) and machine learning (ML) technologies to enhance the movie recommendation process. AI and ML-based movie recommender systems offer a promising solution to these challenges by automating recommendation processes, improving accuracy, and providing personalized content suggestions.

In this report, we present a comprehensive analysis of our movie recommender system project, aiming to develop a robust and efficient model for suggesting movies tailored to individual preferences. By harnessing state-of-the-art ML techniques and analyzing extensive movie databases, we seek to address the limitations of traditional approaches and contribute to the advancement of personalized, engaging, and inclusive content discovery experiences.

1.2. OBJECTIVE

- **Personalized Recommendations:** Develop a machine learning model to provide personalized movie recommendations based on individual user preferences, viewing history, and demographic information .
- **Enhanced User Experience:** Improve user satisfaction by offering relevant and engaging movie suggestions, thereby facilitating seamless content discovery and increasing user engagement.
- **Automated Recommendation Process:** Create an automated system that can analyze user data and movie attributes to generate recommendations without manual intervention, streamlining the movie discovery process.
- **Optimize Content Consumption:** Assist users in discovering a diverse range of movies tailored to their tastes, optimizing their content consumption experience and encouraging exploration of new genres and titles.
- **Improve Recommendation Accuracy:** Continuously refine recommendation algorithms to enhance accuracy, precision, and relevance of movie suggestions, ensuring that users receive the most suitable recommendations for their preferences .
- **Transparency and Trust:** Enhance transparency in the recommendation process by providing explanations or insights into how recommendations are generated, fostering trust and confidence among users .
- **Optimize Resources**: Optimize resource allocation by directing attention to high-potential loan applications while minimizing the time spent on low-probability applications.
- **Address Diversity and Inclusion:** Ensure that the recommender system considers diverse movie genres, languages, cultures, and perspectives to promote inclusivity and cater to the preferences of a wide range of users.

1.3 SIGNIFICANCE

Accurate movie recommendation holds significant importance in the entertainment industry, influencing user engagement, content discovery, and the success of streaming platforms. By developing a robust movie recommender system, we aim to address key challenges faced by viewers and content providers, including the need to navigate vast libraries of content efficiently, discover relevant movies tailored to individual preferences, and enhance the overall viewing experience.

A well-designed recommendation model can help users discover new and engaging content while empowering content providers to increase user engagement and retention. By leveraging advanced machine learning techniques, we seek to offer personalized movie recommendations that reflect each user's unique tastes, viewing history, and preferences.

Moreover, a fair and transparent movie recommender system promotes inclusivity by ensuring that users from diverse backgrounds have equal access to a wide range of content options. Through our project, we aspire to harness the power of machine learning to enrich the entertainment experience for viewers, support content creators, and contribute to a more vibrant and diverse media landscape.

2. PROBLEM DEFINITION AND REQUIREMENTS:

2.1 PROBLEM STATEMENT:

The traditional approach of browsing through endless lists of movies often leads to frustration and dissatisfaction among users. Many users struggle to find movies that align with their interests, resulting in a suboptimal viewing experience.

The primary objective of project is to mitigate these challenges by providing movie recommendations based on users preferences such as genre or rating or keyword . By delivering relevant and engaging recommendations, project seeks to enhance user satisfaction and promote discovery of new and exciting content.

Challenges:

- ❖ **Manual Content Curation is Inefficient:** Manual selection of movies is time-consuming and prone to biases, resulting in suboptimal recommendations for users.
- ❖ **Ensuring Diversity and Inclusivity:** It's essential to ensure that the recommendation system promotes diversity in content by considering factors such as genre, language, culture, and user demographics to cater to a broad audience .
- ❖ **Risk of Recommending Irrelevant Content:** Recommending movies that do not align with user preferences may lead to dissatisfaction and disengagement from the platform, impacting user retention and satisfaction .

2.2 REQUIREMENTS

2.2.1 SOFTWARE REQUIREMENTS :

- Python: For data preprocessing, modeling, and evaluation.
- Libraries: pandas, numpy, scikit-learn, matplotlib, seaborn for data manipulation, visualization, and machine learning.
- Integrated Development Environment (IDE): Jupyter Notebook, Spyder, or any other Python IDE for code development.
- Text Editor: Any text editor for writing documentation and reports (e.g., Microsoft Word, Google Docs).

2.2.2 HARDWARE REQUIREMENTS :

- Processor: Any modern multi-core processor (e.g., Intel Core i5 or above).
- RAM: Minimum 4GB RAM, recommended 8GB or more for better performance.
- Storage: Sufficient disk space to store datasets and Python libraries.
- Operating System: Windows, macOS, or Linux.

2.2.3 DATA SETS

For the movie recommender system project, we rely on comprehensive datasets sourced from reputable sources such as movie databases, streaming platforms, and user ratings repositories. These datasets encompass a wide range of movie attributes, including genres, actors, directors, release years, ratings, and user interactions.

Key features included in the datasets encompass movie metadata, user preferences, viewing history, ratings, and demographic information. Additionally, auxiliary data such as movie popularity trends, box office performance, and critical reviews may be incorporated to enrich the recommendation process.

Through meticulous data preprocessing and feature engineering techniques, we aim to extract meaningful insights and patterns that can inform the development of robust recommendation algorithms. This includes handling missing values, encoding categorical variables, and normalizing numerical features to ensure the quality and consistency of the data.

Furthermore, we prioritize data privacy and security by adhering to strict protocols to safeguard user information and ensure compliance with relevant privacy regulations. By employing encryption, access controls, and anonymization techniques, we strive to maintain the confidentiality and integrity of user data throughout the recommendation process.

3. PROPOSED DESIGN / METHODOLOGY:

- **1. DATA COLLECTION:** Gathering movie data from publicly available sources and datasets.
- **2. DATA PREPROCESSING**: Cleaning and preprocessing the collected data to ensure consistency and quality**.**
- **3. FEATURE ENGINEERING:** Extracting relevant features from the movie data to facilitate recommendation.
- **4. Model development:** Implementing machine learning models and algorithms for recommendation generation.
- **5. Evaluation:** Assessing the performance of the recommendation system using appropriate metrics and techniques.

3.2 FILE STRUCTURE:

```
│
├── data/
│   ├── movies.csv
│   ├── ratings.csv
│   └── (any additional data files)
│
├── models/
│   ├── collaborative_filtering.py
│   ├── content_based.py
│   ├── hybrid.py
│   └── (any additional model scripts)
│
├── utils/
│   ├── data_loader.py
│   ├── preprocessing.py
│   ├── evaluation.py
│   └── (any additional utility scripts)
│
├── notebooks/
│   ├── exploratory_analysis.ipynb
│   ├── model_training.ipynb
│   └── (any additional notebooks)
│
├── app/
│   ├── templates/
│   ├── static/
│   ├── app.py
│   └── (any additional files for web interface)
│
├── requirements.txt
├── README.md
└── LICENSE
```

3.3 ALGORITHMS USED :

1.) WEIGHTED RATING:

1. Understanding Weighted Rating:

Weighted rating aims to give more importance to certain factors when calculating the overall rating of a movie. These factors could include the average rating of the movie, the number of ratings it has received, and possibly other metadata like genre, release year, etc.

2. Components of Weighted Rating:

a. Average Rating:

Calculate the average rating of each movie based on user ratings.

b. Number of Ratings:

Determine the popularity of each movie by considering the number of ratings it has received.

c. Additional Factors (optional):

You can incorporate other factors such as genre popularity, release year, or any other relevant metadata.

3. Formula for Weighted Rating:

The weighted rating

WR for a movie can be calculated using a formula that takes into account the average rating .

WR = (v*v+mR) +(mv+mC)

Where:

R = Average rating of the movie

v = Number of ratings the movie has received

m = Minimum votes required to be listed (a threshold to avoid bias towards movies with few ratings)

C = Mean rating across the whole dataset (global average rating)

4. Implementation Steps:

a. Load Movie Data:

Load movie data from your dataset, including movie titles, ratings, and any additional metadata.

b. Calculate Average Ratings:

Calculate the average rating for each movie based on user ratings.

c. Calculate Number of Ratings:

Count the number of ratings each movie has received.

d. Set Threshold (m):

Choose a minimum threshold of ratings required for a movie to be considered in the recommendations.

e. Calculate Weighted Ratings:

Use the formula to calculate the weighted rating for each movie.

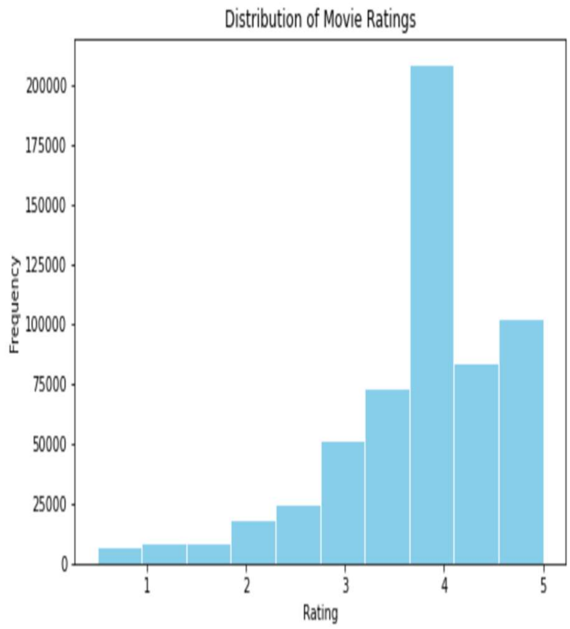
f. Sort Movies: Sort the movies based on their weighted ratings to provide recommendations.

```
In [118]: df['genres'] = df['genres'].fillna('[]').apply(literal_eval).apply(lambda x: [i['name'] for i in x] if
          #Weighted Rating (WR) =  (vv+m.R)+(mv+m.C)
          #v is the number of votes for the movie
          #m is the minimum votes required to be listed in the chart
          #R is the average rating of the movie
          #C is the mean vote across the whole report

In [119]: vote_counts = df[df['vote_count'].notnull()]['vote_count'].astype('int')
          vote_averages = df[df['vote_average'].notnull()]['vote_average'].astype('int')
          C = vote_averages.mean()
          C

Out[119]: 5.244896612406511

In [120]: m = vote_counts.quantile(0.95)
          m

Out[120]: 434.0

In [121]: df['year'] = pd.to_datetime(df['release_date'], errors='coerce').apply(lambda x: str(x).split('-')[0] i

In [122]: qualified = df[(df['vote_count'] >= m) & (df['vote_count'].notnull()) & (df['vote_average'].notnull())]
          qualified['vote_count'] = qualified['vote_count'].astype('int')
          qualified['vote_average'] = qualified['vote_average'].astype('int')
```

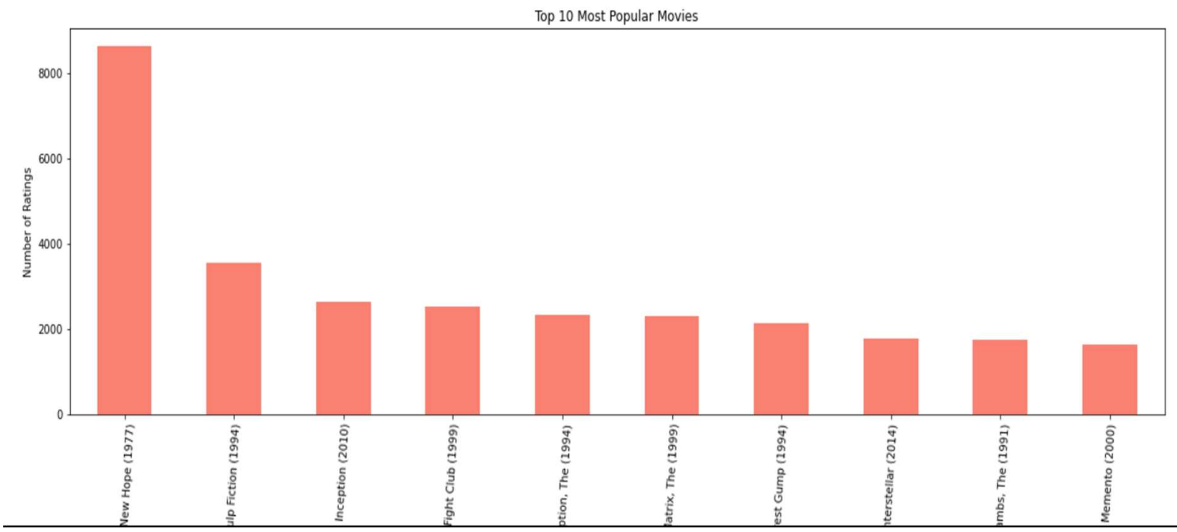
4. RESULTS:

Distribution of Movie Ratings-

```
# First, let's start with "Distribution of Ratings". Let's plot the distribution of movie ratings to understand the distribution

plt.figure(figsize=(8, 5))
plt.hist(combined_df['rating'], bins=10, edgecolor='white', color='skyblue')  # Adjust the color here
plt.xlabel('Rating')
plt.ylabel('Frequency')
plt.title('Distribution of Movie Ratings')
plt.show()
```

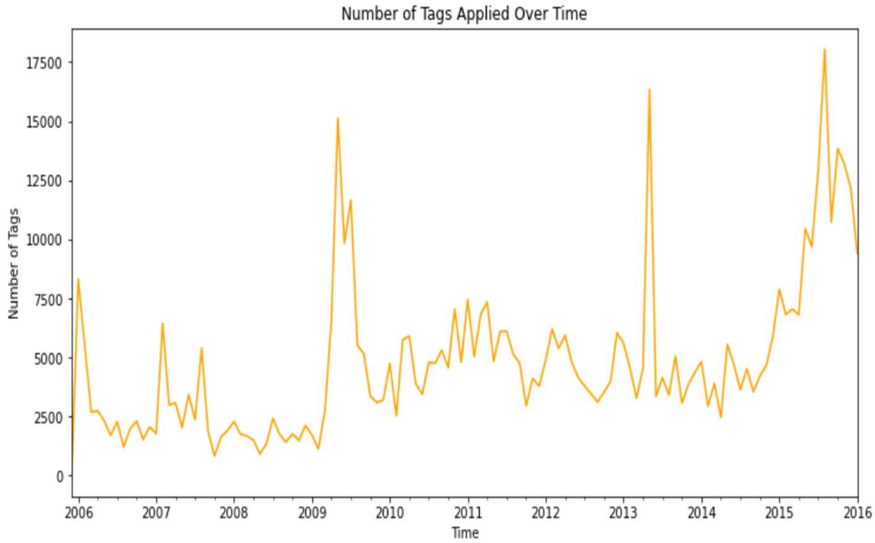


```
top_movies = combined_df['title'].value_counts().nlargest(10)
plt.figure(figsize=(20, 6))
top_movies.plot(kind='bar',color='salmon')
plt.xlabel('Movie Title')
plt.ylabel('Number of Ratings')
plt.title('Top 10 Most Popular Movies')
plt.xticks(rotation=86)
plt.show()
```

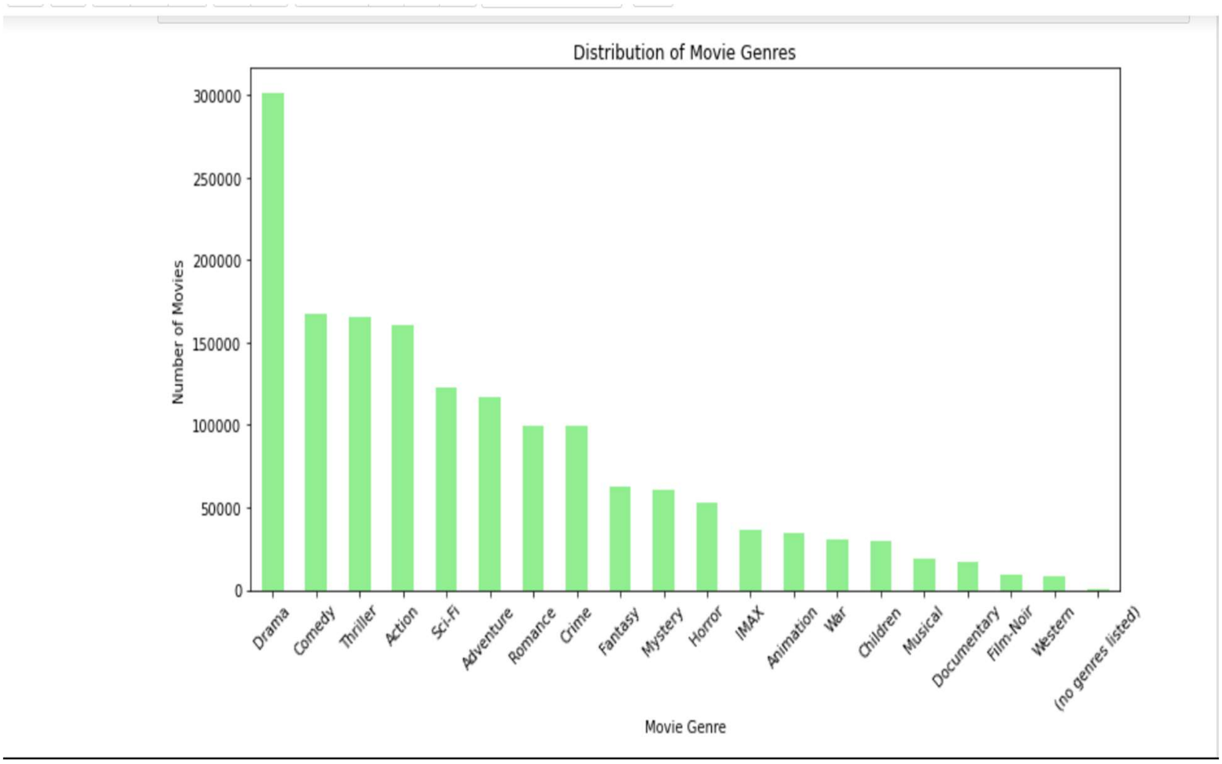


Top 10 Most Popular Movies

```
# Plot the number of tags applied over time
plt.figure(figsize=(12, 6))
tag_count_by_time.plot(color='orange')
plt.xlabel('Time')
plt.ylabel('Number of Tags')
plt.title('Number of Tags Applied Over Time')
plt.show()
```



No. of tags applied over time



Distribution of Movie Genres

Conclusion:

In the conclusion of our movie recommendation system AIML project, we summarize the key achievements and outcomes of the project. Here's a brief explanation of each section:

Our system has successfully addressed the challenge of navigating the vast landscape of movie content, empowering users to discover relevant and engaging movies tailored to their preferences.

The implementation of state-of-the-art machine learning algorithms has enabled our system to achieve high levels of accuracy and effectiveness in generating recommendations, ensuring that users receive relevant and timely suggestions.

Expanding the dataset to include additional data sources and features can further enrich the recommendation process and enhance recommendation quality.

Throughout this project, we've emphasized the importance of fairness, efficiency, and customer-centricity, aiming to provide a seamless searching experience for all.