**Movie Recommender**

A Django Project

*Developed by Rayah Mian*  
Date of Completion: July 30, 2023

Table of Contents

[1. Introduction 2](#_Toc141779004)

[1.1 Objective 2](#_Toc141779005)

[2. Data Collection and Preprocessing 2](#_Toc141779006)

[2.1 Data Collection and Merging 2](#_Toc141779007)

[2.2 Data Cleaning and Transformations 3](#_Toc141779008)

[2.3 Feature Engineering 3](#_Toc141779009)

[3. Exploratory Data Analysis (EDA) 4](#_Toc141779010)

[3.1 Data Overview 4](#_Toc141779011)

[3.2 Key Statistics 5](#_Toc141779012)

[3.3 Data Visualizations 6](#_Toc141779013)

[3.4 Genre Analysis 9](#_Toc141779014)

[3.5 Platform Analysis 11](#_Toc141779015)

[4. Methodology 11](#_Toc141779016)

[4.1 Approach and Algorithms 11](#_Toc141779017)

[4.2 Reasoning Behind Recommendation Technique 12](#_Toc141779018)

[4.3 Implementation 12](#_Toc141779019)

[4.4 Limitations and Challenges 12](#_Toc141779020)

[5. Results and Evaluation 13](#_Toc141779021)

[5.1 Movie Recommender Results 13](#_Toc141779022)

[5.2 Limitations and Future Enhancements 15](#_Toc141779023)

[6. Conclusion 15](#_Toc141779024)

[References 16](#_Toc141779025)

1. Introduction

The Movie Recommender is a web application built using the Django framework, Python, and HTML, aimed to deliver randomized movie recommendations to users based on their given criteria. This project serves as a tool for movie enthusiasts who are seeking to discover new and exciting films that align with their tastes. The foundation of this application lies in its comprehensive dataset of movies. It employs filtering techniques to identify the appropriate movies seamlessly. Throughout this report, we will delve into the development process, data analysis insights, and the technical implementation of the Movie Recommender. If you would like to view all the Jupyter Notebook/Python codes for analysis, visit this link.

1.1 Objective

The primary objective of the movie recommendation system is to simplify the process of movie selection by using the power of data analysis and recommendation algorithms. By collecting user inputs, such as preferred genres, release years, and countries of interest, the application generates a randomized list of three movie suggestions tailored to the user's criteria.

2. Data Collection and Preprocessing

2.1 Data Collection and Merging

For this project, I collected movie datasets for four prominent streaming platforms from Kaggle. These datasets were sourced in CSV format and combined to form a comprehensive movie repository. The individual dataset breakdown is as follows:

1. **Netflix Dataset**: 8,807 entries
2. **Amazon Prime Dataset**: 9,668 entries
3. **Hulu Dataset**: 3,073 entries
4. **Disney Plus Dataset**: 1,450 entries

To create a comprehensive dataset, I merged the four individual datasets into one large dataset using Python and the Pandas library. The final merged dataset, denoted as 'df', contained essential information from each streaming platform and unified view of movie data from all four streaming platforms. It streamlined the information and provided a centralized source for analysis and generating movie recommendations. In total, the merged dataset contained 22,998 entries, each representing an available movie or TV show. The cleaning, transformations, and feature engineering processes led to a refined dataset with 16,103 entries.

The merged dataset consists of the following columns:

1. title: the title of the movie
2. release\_year: the year the movie was released
3. show\_id: a unique identifier for each movie
4. platform: indicates the streaming platform to which the movie belongs
5. type: indicates whether the entry is a movie or a TV show
6. director: the director(s) of the movie
7. country: the country or countries where the movie was produced
8. cast: the cast members starring in the movie.
9. date\_added: the date when the movie was added to the streaming platform
10. rating: the content rating of the movie
11. duration: the duration of the movie
12. genre: the genre(s) of the movie
13. description: a brief description or synopsis of the movie

2.2 Data Cleaning and Transformations

Before proceeding with the analysis, the dataset required cleaning to handle any missing or erroneous data. I performed data cleaning tasks such as removing duplicates, handling null values, and standardizing data types to ensure consistency and usability in subsequent stages. Additionally, since the datasets had both movie and TV show entries, I removed all entries that were of type TV show. The resulting dataset only contained movies, which is important for this application.

2.3 Feature Engineering

Feature engineering played a pivotal role in enhancing the dataset's predictive power and overall performance. The following notable feature engineering techniques were applied to enrich the dataset and improve the movie recommendation process.

1. **Adding the "platform" Column**: To indicate the streaming platform to which each movie belongs, a new column called "platform" was introduced to the dataset. This addition enabled easy identification of a movie's source platform. In cases where identical movie titles existed on different platforms, the entries were merged, and the respective streaming platforms were aggregated within the "platform" column.
2. **Generating Unique Movie IDs**: Since each original dataset had its own unique movie identifiers, a new set of unique movie IDs was generated. This step was important to ensure uniformity across the entire dataset. The new movie IDs replaced the original identifiers in each entry sequentially, simplifying the dataset and promoting seamless integration.
3. **One-Hot Encoding and Mapping**: The dataset underwent one-hot encoding for genre information before applying genre mapping. One-hot encoding converted categorical genre data into binary columns, indicating the presence or absence of specific genres in each movie entry. Following this, a genre mapping dictionary was utilized to group similar genres together, resulting in a more concise and informative representation of movie genres. By consolidating related genres, the dataset achieved more coherence. The same was done for the ratings column.
4. **Release Year Binning**: To simplify the analysis and enhance the interpretability of trends, movie release years were grouped into bins. This binning process allowed for the identification of temporal patterns and trends in movie releases. By aggregating release years into meaningful intervals, the dataset became more accessible for time-based insights.

By employing these preprocessing steps and feature engineering techniques, the dataset was optimized for analysis and recommendation purposes. These transformations paved a personalized movie recommender where users can receive tailored movie suggestions based on their viewing criteria. The subsequent sections will delve into the exploratory data analysis and the utilization of Django and Python to develop the movie recommender application.

3. Exploratory Data Analysis (EDA)

3.1 Data Overview

The movie recommender project utilizes data from four popular streaming platforms: Netflix, Amazon Prime, Hulu, and Disney Plus. To create a comprehensive dataset for analysis, the individual datasets from each platform were merged into one large dataset named 'df', each containing a collection of movies and TV shows with their respective attributes. By merging these datasets, we are creating a comprehensive and consolidated dataset. The purpose of this consolidation is to establish a rich foundation from which I will develop and implement the recommendation system.

A screenshot of a computer

Description automatically generatedA screenshot of a computer code

Description automatically generated

Figure : Before Cleaning Data

Figure : After Cleaning Data

3.2 Key Statistics

A screenshot of a computer

Description automatically generatedThe exploratory data analysis (EDA) of the movie dataset yielded valuable insights into the key characteristics and properties of movies available. Below are the key statistics for the continuous variables: release year and duration.

Figure : EDA on Continuous Variables

**Release Year**:

* The release year is a continuous variable since it involves numerical values that measure a specific point in time on a continuous scale.
* There are 16,103 valid release year entries
* The movies span a range from 1920 to 2021
* The mean release year is approximately 2009
* The standard deviation of the release years is 17.3 which signifies moderate dispersion of movies across the years

**Duration**:

* Duration is a continuous variable since it is represented by numerical values on a continuous scale measured in minutes
* There are 15,714 valid entries
* The minimum duration of a movie is 0 minutes (which may represent missing or incorrect values) and a maximum duration of 601 minutes
* The average duration of a movie is approximately 93 minutes
* The median duration is 94 minutes which serves as a robust measure of central tendency, especially in the presence of outliers
* The standard deviation is 36 minutes
* The interquartile range is from 80 to 109 minutes, covering the middle 50% of the dataset

3.3 Data Visualizations

The visualizations presented in this section are categorized as follows:

1. **Box** **Plots**: To gain insights into the spread and central tendency of release years and durations, we present box plots. These plots offer a concise representation of the distribution's quartiles, median, and potential outliers.
2. **Histograms**: We employ histograms to portray the distribution of release years and movie durations. These graphical representations allow us to discern any patterns in movie releases over the years and explore the range of movie durations in our dataset.
3. **Bar** **Charts**: Bar charts are utilized to visualize the distribution of movies across different streaming platforms and genres. These visualizations provide a clear overview of the popularity of various platforms and genres within our dataset.

By combining these informative visualizations, we aim to present a comprehensive overview of our movie dataset, enabling us to make data-driven decisions and draw meaningful conclusions in the subsequent sections of our analysis.

A graph of a bar graph

Description automatically generated

A graph showing the distribution of release years

Description automatically generated

A graph of a distribution of movie durations

Description automatically generated

A graph of a movie count

Description automatically generated A graph of a movie count

Description automatically generated

3.4 Genre Analysis

In the genre analysis, we delve deeply into the movie genres available in our dataset by exploring their distribution and popularity. Genres play a significant role in shaping users' movie preferences. Through data visualizations, we will gain insights into the most frequent genres, uncover genre trends over time, and identify the top genre combinations that are available.

The following is the top 10 genres and their count percentages:

A list of information on a white background

Description automatically generated

We can analyze genre trends over time. The following are box plots to visualize the distribution of release years for each genre:

A graph with different colored lines

Description automatically generated

To investigate genre distribution by platform, I have compiled a stacked bar chart. This bar chart visualizes movie distributions across the four streaming platforms.

A graph of different colored lines

Description automatically generated

A movie entry may be categorized as one or more genres. We will analyze the top genre combinations. The following are the top 10 genre combinations:

A screen shot of a computer

Description automatically generated

From the genre analysis, we have discovered that the top 10 genres in order are Drama, Comedy, International, Action & Adventure, Thriller, Documentary, Children & Family, Horror, Romantic, and Science Fiction & Fantasy. It is also clear that most genres’ central tendency is spread out from 2000 to 2021 with many outliers. However, Military & War and Western movies’ central tendency ranges from approximately the 1940s to the 2000s with seemingly no outliers. Lastly, we can observe that Netflix has more movie entries for an array of genres.

3.5 Platform Analysis

In the platform analysis, we explore the distribution of movies across different streaming platforms and investigate which platforms have the most diverse selection of movies and which ones have a higher concentration of specific genres.

The platform with the most diverse selection of movies: Action & Adventure

Platform with a higher concentration of specific genres: Drama

4. Methodology

The movie recommender system was built using a content-based filtering approach, which utilizes movie attributes and user preferences to generate randomized recommendations.

4.1 Approach and Algorithms

The approach used in the movie recommender system is content-based filtering, which involves the following steps:

1. User Preferences and Movie Attributes: Users provide their movie preferences by selecting options such as genre, release year, and streaming platform using a user interface. A user can choose multiple options.
2. Filtering Movies based on User Preferences: The system matches user preferences to the available movies in the database and filters out movies that do not meet the selected criteria.
3. Generating Personalized Recommendations: From the filtered movie database, the system randomly selects three movies that meet the user's preferences.

Steps in Creating the Recommendation System:

1. Data Collection and Preprocessing:
   1. Gathered movie datasets from popular streaming platforms: Netflix, Amazon Prime, Hulu, and Disney Plus.
   2. Merged the individual datasets to create a comprehensive movie dataset.
   3. Cleaned and transformed the data, handling duplicates and missing values, and prepared it for analysis.
2. Feature Engineering:
   1. Added a "platform" column to indicate the streaming platform for each movie.
   2. Grouped similar genres together using a genre mapping dictionary for more concise and informative genre representation.
   3. Grouped similar ratings together using a rating mapping dictionary.
   4. Formatted duration values to an integer in minutes.
3. User Interaction and Movie Filtering:
   1. Developed a user interface where users can specify their movie preferences, such as genre, release year, country and streaming platform.
   2. Implemented a movie filtering mechanism to match user preferences with movies in the database and retrieve relevant movie entries.
4. Generating Personalized Recommendations:
   1. Randomly selected a set of movies from the filtered database that align with the user's selected preferences.
   2. Presented the personalized movie recommendations to the user based on their input.

4.2 Reasoning Behind Recommendation Technique

Content-based filtering was chosen for the movie recommender system due to its simplicity and effectiveness in generating personalized recommendations based on user preferences. By allowing users to specify their movie preferences, the system can filter the movie database to provide a selection of movies that match the user's tastes and criteria. This approach is well-suited for applications where users have specific preferences and want to explore movies that align with their interests.

4.3 Implementation

The development process involved coding a Django web application with Python and creating the user interface (UI) using HTML. To build the movie recommender system, we employed the Django web framework, which allowed us to create a dynamic and interactive application. The Python programming language was utilized to implement the backend functionalities and handle user requests. The entire process of development can be summarized in the following steps:

1. Defining the Data Model: We designed the data model to store movie-related information such as title, release year, genre, duration, platform, and other relevant details. The data model facilitated efficient storage and retrieval of movie data from the database.
2. Data Collection and Preprocessing
3. Feature Engineering and Mapping
4. Implementing the Recommendation Algorithm
5. Designing the User Interface: The user interface was created using HTML, providing an intuitive and user-friendly platform for users to input their movie preferences.

4.4 Limitations and Challenges

During the implementation process, several challenges were encountered due to the richness of the movie data:

1. Genre and Platform Diversity: Movies can belong to multiple genres and be available on different streaming platforms, leading to data entries with multiple values. Extracting and manipulating this information required careful handling to ensure accurate recommendations.
2. Data Quality and Missing Values: The quality and completeness of the movie datasets posed challenges. Some entries had missing values or inconsistent data, necessitating data cleaning and preprocessing to ensure reliable recommendations.
3. Genre Mapping Accuracy: Creating an accurate genre mapping dictionary required extensive research and analysis. Mapping similar genres together could sometimes be subjective, impacting the effectiveness of genre-based recommendations.

5. Results and Evaluation

In this section, we present the results of the movie recommender system, limitations and future enhancements.

5.1 Movie Recommender Results

The movie recommender system successfully delivers randomized movie recommendations based on user preferences for genre, platform, release year and country. After entering their desired criteria, users receive three movie suggestions that match their input. The recommendations are displayed on the results page, providing users with a concise overview of the suggested movies' titles, rating, duration, and description. Here are screenshots of the application’s input form and a sample result:

A screenshot of a movie generator

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

5.2 Limitations and Future Enhancements

While the movie recommender system demonstrates considerable promise, it is important to acknowledge its limitations and areas for future enhancements:

1. Limited User Data: The system primarily relies on user input to generate recommendations. Enhancements could involve incorporating user interaction data, such as movie ratings and watch history, to refine the recommendations further.
2. Handling New Movies: As new movies are released, they are not included in the database. This database only includes movies from 1920 to 2021. Introducing a mechanism to periodically update the movie dataset would ensure that users receive recommendations for the latest releases.
3. Fine-tuning Genre Mapping: The accuracy of genre mapping impacts the relevance of recommendations. Refining the genre mapping dictionary based on user domain expertise can lead to more accurate genre-based suggestions.
4. User-Based Recommendations: Incorporating a user database and keeping track of users’ likes and dislikes may be essential for future algorithms that provide more personalized recommendations.

6. Conclusion

In conclusion, the Movie Recommender project successfully developed an interactive system that delivers randomized movie recommendations. The system's content-based filtering approach, coupled with feature engineering techniques, contributed to relevant and diverse movie suggestions. While the project achieved its primary objectives, continuous improvements and enhancements can further refine the movie recommender system to ensure optimal performance and user satisfaction.

References

1. Amazon Prime Movies and TV Shows Dataset:

* Kaggle. Amazon prime movies and TV shows. Retrieved from <https://www.kaggle.com/datasets/shivamb/amazon-prime-movies-and-tv-shows>

1. Disney Movies and TV Shows Dataset:

* Kaggle. Disney movies and TV shows. Retrieved from <https://www.kaggle.com/datasets/shivamb/disney-movies-and-tv-shows>

1. Hulu Movies and TV Shows Dataset:

* Kaggle. Hulu movies and TV shows. Retrieved from <https://www.kaggle.com/datasets/shivamb/hulu-movies-and-tv-shows>

1. Netflix Shows Dataset:

* Kaggle. Netflix shows. Retrieved from <https://www.kaggle.com/datasets/shivamb/netflix-shows>