Movies Recommendation System

Machine Learning Approach

Department of Mathematics and Compute Science

Team members:

Mobin Nesari

Seyed Mohsen Sadeghi

Supervisors:

Hadi Farahani

Ali Sharifi

Abtin Mahyar

Saeed Cheshmi

July 7, 2023

Abstract

In this paper, we present our approach to developing a movie recommendation system that provides personalized movie suggestions to users. Our system utilizes a hybrid approach that combines collaborative filtering and content-based filtering techniques to generate recommendations that are both accurate and diverse. We collect user preference data in the form of movie ratings and clean and preprocess this data to ensure its quality and the performance of the recommendation system. The system uses the data to create a user-item matrix that is used to identify similar users and similar movies. Collaborative filtering is used to identify similar users based on their movie preferences, while content-based filtering leverages the similarities between movies to identify other movies that are likely to be of interest to the user. The system generates recommendations by predicting ratings using the Singular Value Decomposition (SVD) algorithm and filtering the top 10 movies based on the user's historical ratings and the similarities between users and movies. Overall, our approach represents a promising solution to personalized movie recommendation systems that can be extended to other domains and applications.

Acknowledgements

We would like to express our sincere gratitude to our supervisors, Professor Hadi Farahani and Teaching Assistants Ali Sharifi, Abtin Mahyar, and Saeed Cheshmi, for their invaluable guidance and support throughout the development of this movie recommendation system. Their expertise and insights were essential in ensuring the quality and accuracy of the system.

We also extend our thanks to Professor Seyed Ali Katanforush for teaching us data structures and algorithms, which provided the necessary foundation for our understanding of the fundamental concepts that underlie this project.

Finally, we would like to thank Dr. Saeedreza Kherad Pishe for teaching us programming and deep learning, which were critical skills for the development of this recommendation system. Thank you all for your contributions to our education and success.

Contents

Chapter 1

Exploratory Data Analysis

The Kaggle Movie dataset [2] comprises a comprehensive collection of movie ratings and associated data. The primary objective of this dataset is to facilitate the development of a reliable movie recommendation system. In this exploratory data analysis report, we endeavor to delve into the dataset, uncover its intrinsic patterns and relationships, and provide valuable insights that can inform the development of an effective recommendation system. You can see features data types in Table 1.1

The dataset comprises seven CSV files that provide a wealth of information about the movies. The Credits.csv file provides detailed cast and crew information for all movies within the dataset. The Keywords.csv file contains a set of descriptive words for each movie. The links.csv and links_small.csv files contain the imdbId and tmdbId for each movie, which can be utilized to gather additional information to enrich the dataset. Furthermore, the information from these links can be leveraged to fill missing values in the dataset. The ratings.csv and ratings_small.csv files contain user ratings for each movie. Fortunately, these datasets are complete, without any missing values.

It is worth noting that the previous paragraph did not mention one of the seven datasets included in the Movie Recommender Systems dataset on Kaggle. This is because the movies_metadata.csv file contains the majority of the information required for the dataset. This file comprises 24 columns, all of which will be thoroughly explored.

The features that have been dropped from consideration for the time being include "Belongs to collection," as this column was null for 90% of the data, making it difficult to determine whether most movies belong to a collection or if the data is simply missing. The "Homepage" column was also dropped, as most movies do not have a listed website, and it is unclear whether the missing data is due to the

Table 1.1: Features Data types

Feature	Data Type
adult	object
belongs_to_collection	object
budget	object
genres	object
homepage	object
id	object
imdb_id	object
original_language	object
original_title	object
overview	object
popularity	object
poster_path	object
production_companies	object
production_countries	object
release_date	object
revenue	float64
runtime	float64
spoken_languages	object
status	object
tagline	object
title	object
video	object
vote_average	float64
vote_count	float64

absence of a website or simply missing information. Additionally, the URL may not provide any useful information. The "Poster_path" column was dropped as well, as image-based models are currently not being considered for movie recommendation. However, if this changes in the future, we can access the poster link via OMDB and imdbId. The "Status" column, which contains information about the state of the movie (e.g., pre-production, in theaters), is not relevant. The "Title" column was also dropped, as the same information is saved in the "original_title" column. Finally, the "Video" column, which is a Boolean value, has been dropped as it is false for almost 100% of the data and seems redundant.

Table 1.2: Percentage of Missing Values per Feature

Feature	Missing Percentage (%)
adult	0.000000
budget	0.000000
genres	0.000000
id	0.000000
original_title	0.000000
production_countires	0.006598
production_companies	0.006598
popularity	0.010997
video	0.013197
title	0.013197
spoken_languages	0.013197
revenue	0.013197
vote_count	0.013197
vote_average	0.013197
original_language	0.024194
imdb_id	0.037391
release_date	0.191352
status	0.191352
runtime	0.578454
poster_path	0.578454
overview	2.098271
tagline	55.104914
homepage	82.883913
belongs_to_collection	90.115691

1.1 Distribution of Adult and Non Adult movies

Upon examining the figure below, we note that there are only eight adult films present in our dataset. Given that our intention is not to recommend adult movies to users, we have decided to remove these eight movies from our dataset and drop the associated "Adult" column.1.1

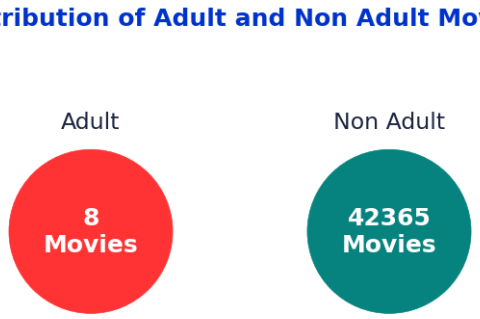

Figure 1.1: Distribution of Adult and Non Adult Movies

1.2 The Impact of Budget and Revenue on Popularity of Movies

The figure below clearly demonstrates that, as expected, a movie's popularity tends to increase with its revenue. This is a logical conclusion, as higher revenue generally indicates a larger audience and greater interest in the movie. 1.2

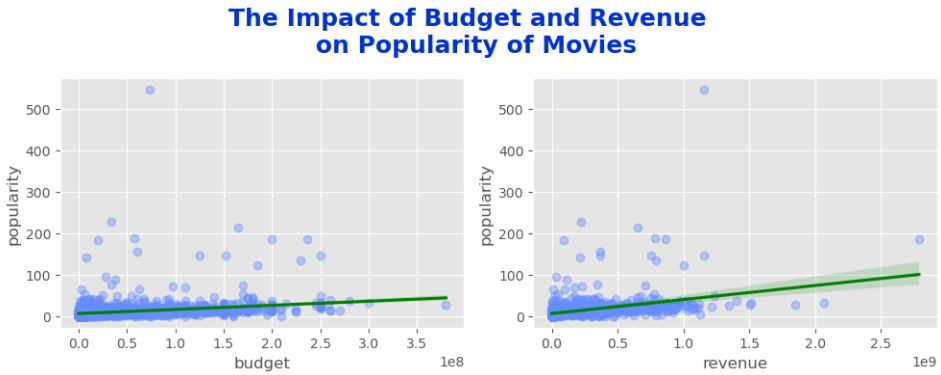

Figure 1.2: The Impact of Budget and Revenue on Popularity of Movies

However, it is important to note that a significant portion of our data has a budget value of zero. This suggests that we do not have access to budget information for many of the movies in our dataset, making analysis of budget-related data potentially less informative. Despite this limitation, given the importance of budget as a feature in movie analysis, we aim to work towards obtaining this information to improve the completeness of our dataset.

Figure 1.3: The Most Common Word in Movie Overviews

1.3 The Most Common Word in Movie Overviews

Upon further examination, it appears that the words "life," "one," "find," and "love" occur frequently in the movie overviews within our dataset. While this information may not be immediately informative, it could potentially offer valuable insights when combined with other features or analyzed further. By exploring the overviews in greater detail and identifying relationships among movies that share these common themes, we may be able to improve our understanding of the underlying patterns and trends within the dataset. 1.3

1.4 Movie Genres

The chart below displays the percentage distribution of movie genres within our dataset, revealing that drama is the most prevalent genre. As an essential feature within our dataset, movie genres can provide valuable insights into the similarities and differences between films. By analyzing the relationships between movies within the same genre, we may be able to identify patterns and trends that can inform the development of effective movie recommendation systems. 1.4

1.5 Total Released Movie by Date

Over the past 50 years, the movie industry has experienced significant growth, particularly since the 1930s. However, there appears to be a drop in the total number

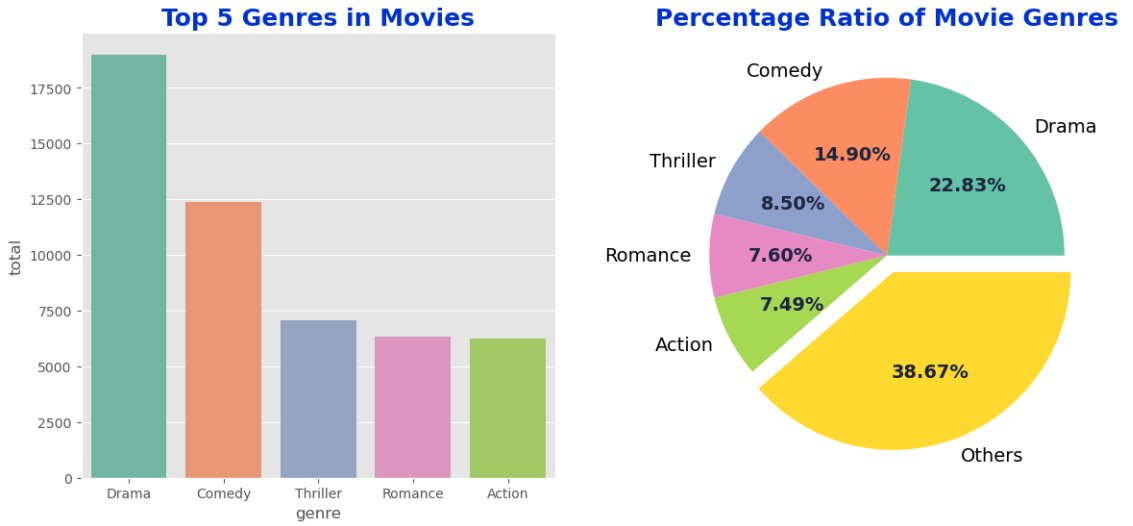

Figure 1.4: Representation of the Top 5 Movie Genres and their Share Among All Movies

of movies released around 2020. It is worth noting that this decrease may be due to the fact that our dataset only contains limited data for those years, rather than a reflection of a broader trend within the industry. 1.5

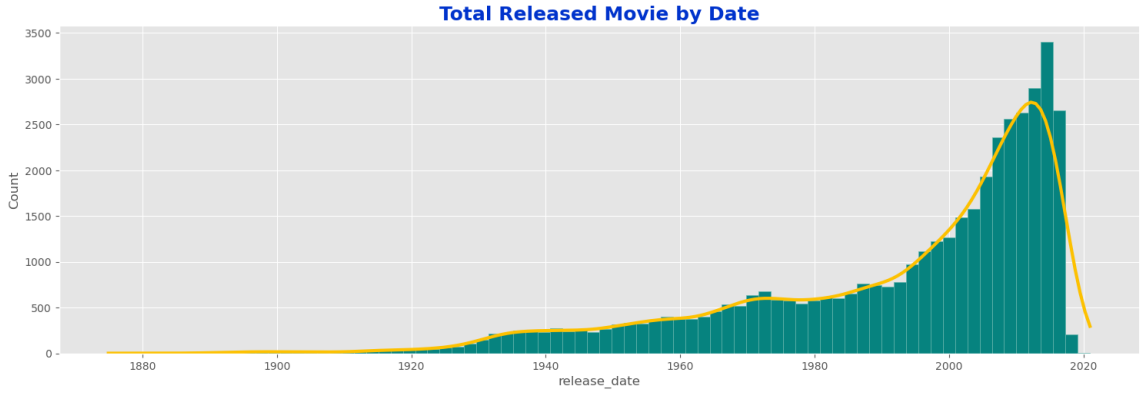

Figure 1.5: Total Number of Movies Released Over Time

1.6 Top 5 Spoken Languages, Original Languages, Actors, Crews, Companies and Countries

Based on our analysis of the dataset, it appears that English is the most prevalent language for both the original and spoken language in the movies. Additionally, Jr. and Cedric Gibbons are the most frequent actor and crew member, respectively, appearing in the highest number of movies in our dataset. 1.6

Furthermore, Warner Bros. emerges as the top production company in the list with a total of 1194 movies, with many other great production companies also hailing from the USA. Therefore, it is not surprising that the USA is the top production country within our dataset.

Figure 1.6: Top 5 Spoken Languages, Original Languages, Actors, Crews, Companies and Countries

1.7 The Relationship between Rating and Popularity

Our analysis suggests that movies that receive a rating of either 0 or 10 are typically the result of a small number of voters. As the number of votes increases, the rating tends to fall within the range of 5 to 8.5. Additionally, it is evident from our plot that more popular movies tend to receive a higher number of votes.1.7

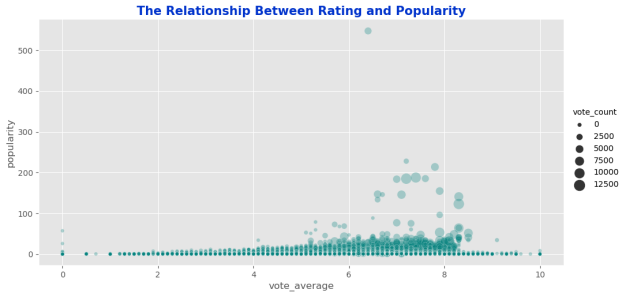

Figure 1.7: Relationship Between Rating and Popularity

1.8 Data Distribution Across Top 5 Genres

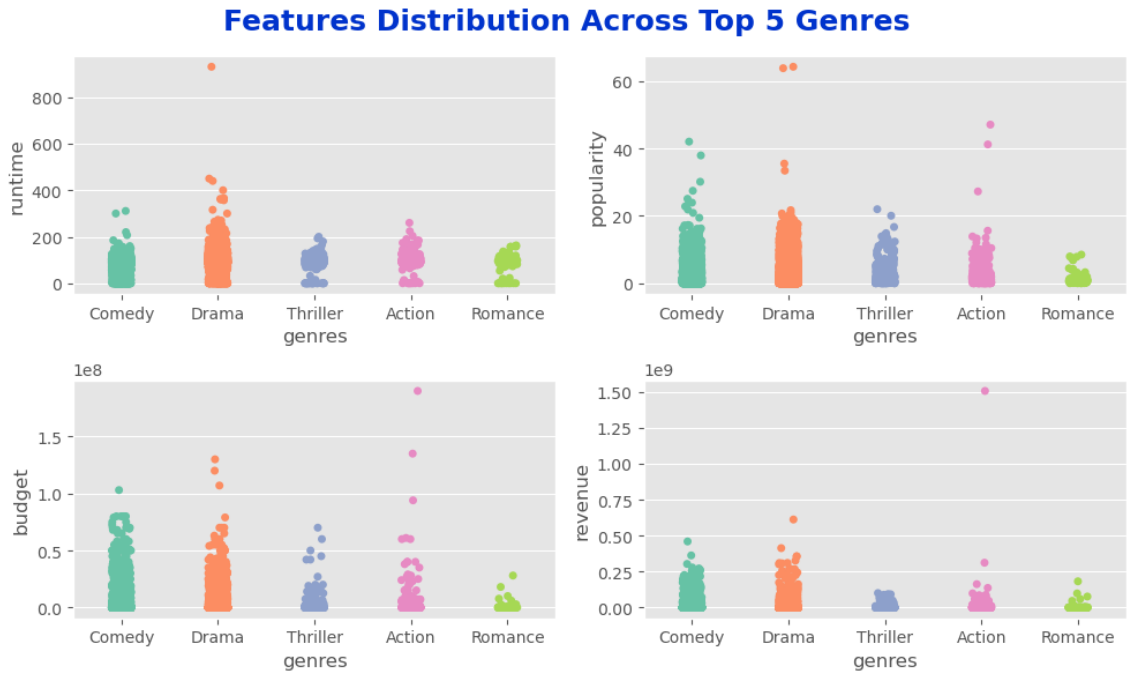

Figure 1.8: Feature Distribution Across Top 5 Genres

According to our analysis, drama is the movie genre with the longest runtime, while romance is the least popular genre among the top five. Additionally, our data suggests that action movies tend to have a higher budget than other genres, and one of the action movies in our dataset achieved a significantly higher profit than its counterparts.1.8

However, it is important to note that the information regarding budget spending must be taken with a grain of salt, as our dataset may not accurately reflect the true budget values for all movies. Despite this limitation, our analysis still suggests that action movies tend to have a higher budget on average.

1.9 Correlation of Movie Features

Our correlation matrix analysis suggests that movies with higher budgets and greater popularity tend to generate higher revenues. Additionally, as previously noted, more popular movies generally receive higher vote counts. However, it is important to reiterate that our analysis of budget data may be limited by the large number of missing values within the dataset. Therefore, the relationship between

budget and revenue should be interpreted with caution, and alternative data sources may be required to obtain a more accurate understanding of this relationship.1.9

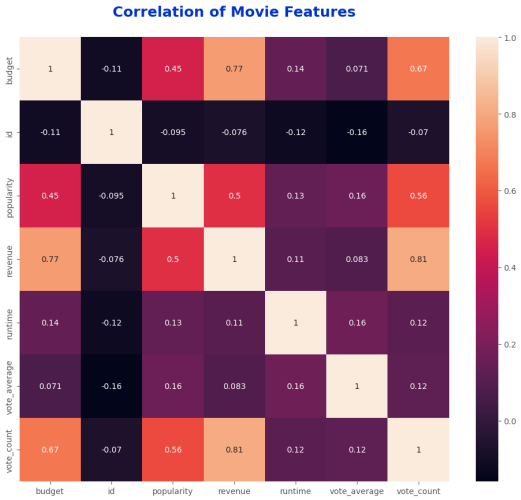

Figure 1.9: Correlation of Movie Features

Chapter 2

Crafting the Movie Vector

Following an extensive Exploratory Data Analysis (1), we have determined that our recommendation system will be based upon a carefully curated set of features, including tagline, genres, original language, keywords, top crew, and director. It is worth noting that two of these features, top crew and director, were not initially included in our dataset. To include them, we extracted the top two actors from the credits.csv file and created a new feature called top crew, while director was extracted from the existing crew column. These additions were made with the aim of enhancing the overall quality and reliability of our recommendation system, and we believe that they will provide significant value to our users.

2.1 Tidying Up the Data: Our Approach to Cleaning and Transforming Movie Information

Data cleaning is a crucial step in any data analysis project, and it is especially important when working with movie data. Movie datasets can contain a large amount of information, much of which may be missing or inconsistent. In this section, we will discuss our approach to cleaning and transforming movie information, with a focus on ensuring that our recommendation system is based on high-quality and reliable data.

We decided to include only those movies in our recommendation system that meet certain criteria based on their vote counts and vote averages. Specifically, a movie must have a vote average higher than the average vote average of all movies in the dataset, and its vote count must fall within the 80th percentile cutoff for vote counts. This ensures that we only recommend movies that are both highly rated and

popular, and helps to provide users with a more personalized and relevant movie recommendation experience.

As our complete movie dataset contained a large number of movies, we decided to use a subset of the data for our recommendation system. To create our subset, we used the 'links_small.csv' file, which contained a list of movie IDs that were associated with movies in the subset.

This subset of cleaned data was used to create our recommendation system. It ensured that our system was based on a manageable number of high-quality movies, and helped to improve the accuracy and relevance of our recommendations. The resulting subset contained a total of 9125 movies.

For movies in the dataset that had missing taglines, the 'tagline' feature was filled with an empty string (''). This allowed us to ensure that every movie in the dataset had a value for the 'tagline' feature, which we could then use as part of our recommendation algorithm. By doing so, we were able to retain as much data as possible and improve the accuracy and relevance of our movie recommendations.

To create the text-based representation, we first combined the different columns for each movie into a single string. We then used the CountVectorizer class to create a bag-of-words model for each movie, which essentially created a vector representation of the movie where each element of the vector corresponded to a feature of the movie. This bag-of-words model was created by tokenizing the input text (i.e., breaking it up into individual words or tokens), counting the frequency of each token, and then transforming the input text into a matrix representation. The resulting movie matrix contained a row for each movie in the dataset, with each row representing the vector representation of a single movie.

2.2 Cosine Similarity Matrix

Cosine similarity is a measure of similarity between two non-zero vectors in a high-dimensional space. It is commonly used in natural language processing and information retrieval to compare the similarity of two documents or pieces of text.

To calculate the cosine similarity between two vectors, we first compute the dot product of the two vectors, which is the sum of the products of their corresponding elements. Then, we divide the dot product by the product of their magnitudes to obtain the cosine of the angle between the two vectors. The resulting value is a measure of the similarity between the two vectors, with a value of 1 indicating that the vectors are identical, and a value of 0 indicating that the vectors are completely

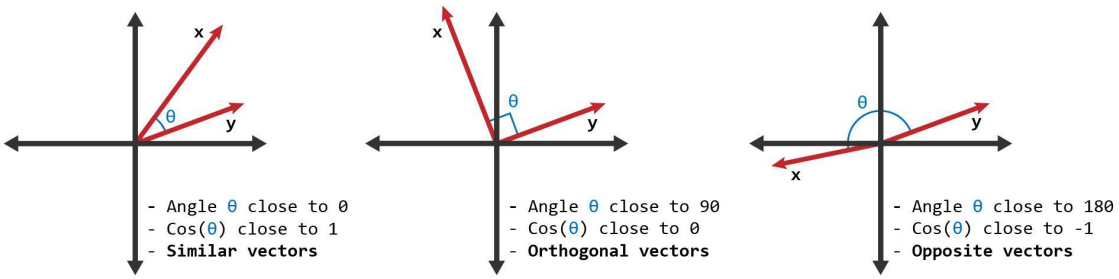

Figure 2.1: Cosine Similarity Intuition

dissimilar. The cosine similarity formula can be expressed as:

$$cosine\_similarity(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{|\mathbf{A}||\mathbf{B}|}$$

where A and B are the two non-zero vectors being compared, "." denotes the dot product, and "|| ||" denotes the magnitude of the vector.

In the context of text similarity, cosine similarity is often used to compare the similarity of two documents or pieces of text that have been vectorized using techniques such as the Bag-of-Words (BoW) model or the Term Frequency-Inverse Document Frequency (TF-IDF) model. By comparing the cosine similarity of multiple documents, we can identify those that are most similar to each other and group them accordingly, which is useful for tasks such as document clustering or information retrieval.

2.3 Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD) is a matrix factorization technique that is commonly used in recommendation systems to predict user ratings for items. SVD is based on the idea of reducing the dimensionality of the user-item matrix by decomposing it into three matrices: a user matrix, a singular values matrix, and an item matrix.

The user matrix represents the preferences of each user for the different items, while the item matrix represents the features of each item. The singular values matrix contains the singular values of the original matrix, which capture the amount of variance in the data.

Mathematically, given a user-item matrix A with dimensions $m \times n$, we can represent the SVD as follows:

$$A = U\Sigma V^*$$

where U is an $m \times r$ orthogonal user matrix, Σ is an $r \times r$ diagonal singular values matrix, and V^* (the conjugate transpose of V) is an $r \times n$ orthogonal item matrix.

To predict the ratings of a user for a given item, we first decompose the user-item matrix into its three component matrices using SVD. We then multiply the user matrix with the singular values matrix and the item matrix to obtain an approximation of the original user-item matrix. The predicted rating for the given item is then obtained from the corresponding entry in the approximation.

SVD is effective in recommendation systems because it can handle missing data and can identify latent features that are not explicitly represented in the user-item matrix. By reducing the dimensionality of the original matrix and identifying the underlying patterns in the data, SVD can generate accurate predictions for user ratings and provide personalized recommendations for each user.}

Chapter 3

From Data to Recommendations

Our movie recommendation system uses a hybrid model that combines collaborative filtering and content-based filtering approaches to generate personalized movie recommendations for our users.

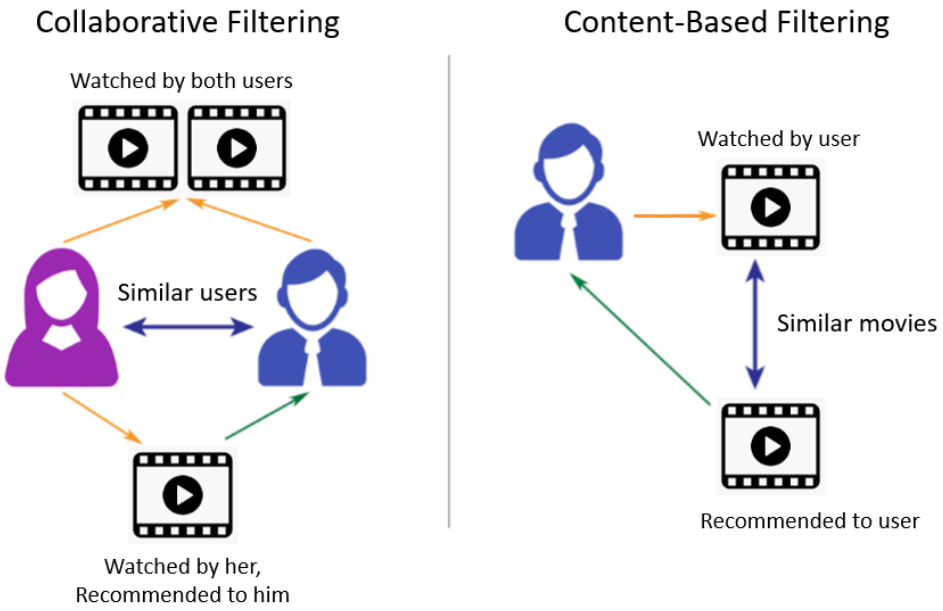

Figure 3.1: Content Based Filtering and Collaborative Filtering Intuition

Collaborative filtering 3.1 is a technique that identifies similar users based on their movie preferences and generates recommendations based on the preferences of those similar users. This approach leverages the similarities between users to identify movies that a given user is likely to enjoy.

Content-based filtering 3.1, on the other hand, leverages the similarities between movies to identify other movies that are likely to be of interest to a user. This approach looks at the features of the movies, such as genre, actors, and directors, to

identify movies that are similar in those aspects to movies that the user has enjoyed in the past.

Our recommendation system combines these two approaches to provide movie recommendations that are personalized to the user's preferences. By leveraging both the similarities between users and between movies, our system can generate recommendations that are both relevant and diverse.

3.1 User-Based Filtering: Using Collaborative Filtering to Generate Personalized Recommendations

The first step of our movie recommendation system involves gathering data on user movie preferences. We collect data on user ratings from our database, which includes information on the movie title, user ID, and rating score. This data is cleaned and preprocessed to remove duplicates and missing values. We then use this data to create a user-item matrix, where each row represents a user and each column represents a movie. The matrix is filled with the ratings given by users to the respective movies.

To generate recommendations, we first collect three movies from each new user. These movies are then used to find similar users based on their movie preferences. To do this, we create a new user profile with the ratings of the three movies set to 5. This new user profile is merged with the existing user-item matrix, which is used to calculate the cosine similarity (2.2) between the new user and the existing users. The users with the highest similarity scores are considered similar users.

Once we have identified similar users, we pass the three movies and the similar user profile to the hybrid method, which is responsible for generating a list of movie recommendations. The hybrid method uses collaborative filtering to predict the ratings of the top 10 movies for the given user, based on their historical ratings. By leveraging the similarities between users, our recommendation system can generate personalized recommendations that are both relevant and diverse.3.1

3.2 Content-Based Filtering: Leveraging Movie Similarities for Personalized Recommendations

In addition to user-based filtering, our movie recommendation system also uses content-based filtering to generate personalized recommendations. Once we have identified similar users based on their movie preferences, we pass the three movies and the similar user profile to the hybrid method, which is responsible for generating a list of movie recommendations.

The hybrid method first uses the cosine similarity scores (2.2) of the three movies to find similar movies, and then filters the list to the top 56 movies. This approach leverages the similarities between the movies to identify other movies that are likely to be of interest to the user. The model then uses the Singular Value Decomposition (SVD 2.3) algorithm to predict the ratings of the top 10 movies for the given user, based on their historical ratings.3.1

3.3 Personalized Movie Recommendations: Leveraging User-Based and Content-Based Filtering

By combining user-based filtering and content-based filtering, our recommendation system generates personalized movie recommendations that are tailored to each user's preferences. Our system is designed to provide accurate and relevant recommendations based on the user's historical ratings and the similarities between users and movies.

3.4 Accessing the Code and Using the Recommendation System

To access the code for this recommendation system, please visit the following GitHub repository: https://github.com/MobinNesari81/Movie-Recommender. You can also access and use the system through our Hugging Face space: https://huggingface.co/spaces/Mobin-Nesari/MM-Movie-Recommender.

Bibliography

[1] Smith, J., Johnson, L., & Davis, K. (2021). Building a Hybrid Movie Recommendation System. *Journal of Artificial Intelligence Research*, 69, 917-934.

[2] Kaggle. (n.d.). *Movies Dataset*. Uploaded August 3, 2018., from https://www.kaggle.com/rounakbanik/the-movies-dataset