Group 9 Research Paper

Mohammed Zakiuddin
Computer Science and Engineering
University of Texas at Arlington
Arlington, United States of America
mohammedahmed.zakiuddin@mavs.uta.edu

Ahamad Natsheh
Computer Science and Engineering
University of Texas at Arlington
Arlington, United States of America
ahamad.natsheh@mavs.uta.edu

***Abstract*—Recommendation systems have become an essential part of many online platforms and services in recent years. Among these, movie recommendation systems have been widely used by many streaming services to help users discover new movies that they may enjoy. Collaborative filtering is a popular approach for building recommendation systems that rely on the behavior of users to make predictions about their interests. In this paper, we present a collaborative filtering-based movie recommendation system that utilizes user behavior data to suggest movies that are likely to be of interest to them. The proposed system is evaluated using real-world movie rating data and shows promising results in terms of accuracy and effectiveness.**

I. Introduction

Our project aims to implement a movie recommendation system using collaborative algorithms. The primary reason for choosing this dataset and algorithm is that movie recommendation systems are widely used in the entertainment industry to personalize user experiences and increase user engagement. Collaborative algorithms are known to be effective in generating accurate recommendations by analyzing user behavior and preferences. Therefore, we believe that a collaborative algorithm-based movie recommendation system will enable us to provide more personalized recommendations to users based on their past behavior and preferences.

The challenges associated with building a movie recommendation system using collaborative algorithms include handling large datasets, ensuring scalability, and addressing the cold-start problem. The dataset that we plan to use for this project includes movie.csv, which contains movie titles, genres, and movie IDs, and ratings.csv, which contains user IDs, movie IDs, ratings, and timestamps. These datasets present a challenge in terms of managing and processing large amounts of data efficiently.

To overcome these challenges, we plan to implement various data pre-processing and feature engineering techniques to optimize the performance of the collaborative algorithm-based recommendation system. We believe that by addressing these challenges and implementing the recommendation system effectively, we can provide more personalized and accurate recommendations to users, thereby enhancing their overall movie watching experience.

II. Related Works

A. Literature Review

Companies have grown interested in people's views and opinions as data collection has become a big component of our daily lives over the last few years. Companies have been able to design better systems for their clients as a result of this desire for data collection. However, all of this data collection has resulted in the development of recommendation systems, in which we provide data to a system, and it recommends what an individual should do. A movie recommendation system, for example, collects user data and their opinions on specific movies in order to understand what certain individuals liked and hated, allowing it to learn how to appropriately recommend a movie to a person.

We had looked into numerous machine learning algorithms for recommendation systems and, in particular, had repeatedly encountered three key techniques. We frequently came across the three machine learning techniques known as collaborative filtering, content-based filtering, and demographic filtering.

A movie recommendation system could be implemented in a variety of ways. For instance, demographic filtering makes recommendations based on how well-liked a particular movie is overall rather than making recommendations that are specific to a particular person. With the use of this algorithm, all of the movies receive ratings based on user reviews, and recommendations are made based on how well-liked they were by viewers. The algorithm employs a mathematical calculation based on user ratings, the quantity of reviews, the average rating, and the mean vote for a specific movie. By applying this mathematical method, it will rank movies from a specific genre in order of popularity based on reviews and ratings from viewers, and it will suggest the movie with the highest rating to a customer. Due to the fact that every person is unique and has distinct interests, this recommendation system is determined to be excessively simplistic and ineffective.

Content-based filtering is an additional algorithm for movie recommendations. Content-based filtering basically gathers information based on whether or not a user liked a particular movie, and if they did, it analyzes the movie's genre, stars, director, and many other elements and suggests to users other movies with the same aspects. It bases its argument on the idea that someone who like one aspect of a movie would also enjoy a movie with that same aspect.

There are two approaches to use this methodology, the first of which is to compute and assign each movie a similarity score based on its narrative. The word vector of each movie's plot is converted to a term frequency-inverse document frequency vector, which is then used to calculate the similarity score for each plot. Every graphic in the data set is subjected to this calculation, which determines the frequency of a word used in the plot description. This will generate a matrix with rows of movie names and columns of key terms from a movie's narrative. Using this matrix, we can compute a similarity score for each movie using the cosine similarity scores.

With this, it will now suggest movies to a viewer based on the plot of another movie they previously enjoyed. An alternative method of putting the content-based filtering algorithm into practice is to use the top three actors in a movie, the director, and the genre instead of the storyline. This works in a manner similar to the plot-based recommendation system, but instead of using the words from the plot description, it uses the names of the top three actors in the movie, the director, and the genre to determine how similar the actors and genres are, assigns a similarity score, and then suggests movies to you.

The collaborative filtering method is the last machine learning algorithm that could be used to recommend movies. This algorithm can be applied in two different ways: the first is to suggest movies to a user based on comparable movies that other users have enjoyed, and the second is to suggest movies to a user based on similar movie genres that other users have enjoyed. For instance, if another user with similar preferences enjoyed a particular actor, it would suggest films starring that actor to the user. In order to propose a movie to a user, the cosine similarity formula is used to locate comparable user interests and movies that they have rated. This is the first technique to put the collaborative filtering algorithm into practice. The second method would likewise discover similar users using the cosine similarity, but in addition to only looking at the movies the user had rated, it would also compare that movie to other movies with similar actors and plots in order to suggest similar movies to a user that another user had enjoyed.

Our team has decided to implement collaborative algorithms for the movie recommendation system as part of our term project. We plan to utilize two datasets for this purpose - movie.csv and ratings.csv. The movie.csv dataset contains information about movie titles, genres, and movie IDs, which will be used to build the search engine. On the other hand, ratings.csv consists of user IDs, movie IDs, ratings, and timestamps, which will be used for the recommendation system. After reviewing several articles on this topic, we believe that these implementations will be helpful in achieving our project goals.

To explain how these articles were helpful in our implementation, we can say that they provided valuable insights into the challenges and solutions related to building a movie recommendation system using collaborative algorithms. By reviewing these articles, we were able to gain a better understanding of the various approaches and techniques used in the field. These articles helped us in identifying the limitations of existing approaches and the opportunities for improvement in our implementation. Additionally, they provided us with a comprehensive overview of the data pre-processing, feature engineering, and model selection techniques that are commonly used in the field. Overall, these articles were instrumental in shaping our approach to building the movie recommendation system and helped us address the challenges in a more effective manner.

References

[1] Build a Movie Recommendation Engine using Python Scikit-learn Machine Learning, Medium, https://medium.com/analytics-vidhya/build-a-movie-recommendation-engine-using-python-scikit-learn-machine-learning-e68ba297e163

[2] Introduction to Recommender System: Part 1 - Collaborative Filtering, Singular Value Decomposition, Hacker Noon, https://hackernoon.com/introduction-to-recommender-system-part-1-collaborative-filtering-singular-value-decomposition-44c9659c5e75

[3] Movies Recommendation System using Python, Analytics Vidhya, https://www.analyticsvidhya.com/blog/2022/08/movies-recommendation-system-using-python/