Recommendation System for Movies

Ze Li (zl2746)

zl2746@cumc.columbia.edu

1 Introduction

A new era of information has been brought about by the quick expansion of data collection. Recommendation systems exist precisely because data is being used to build more effective systems. By improving the quality of search results and providing items that are more appropriate to what the user is looking for or that are consistent with their search history, a recommendation system serves as an information filter. These programs forecast a user's preference or rating for a certain product. Many large digital companies utilize this technology in different ways: YouTube chooses which video to show next, Amazon suggests things to users, and Netflix and Spotify rely extensively on recommendation engines to power their services and operations.

The dataset used in this project comprises metadata for 45,000 movies from the Full MovieLens dataset, covering releases up to July 2017. A wide range of data is included, such as actors, crew, keywords, budgets, earnings, posters, release dates, languages, production firms, nations, and ratings and votes on TMDB. Furthermore, the data set offers user ratings for both the whole dataset with 26 million ratings from 270,000 people and a smaller subset of 9,000 movies, ID mappings between TMDB and IMDB, and JSON-formatted data for keywords and credits. These extensive data sources offer a solid foundation for investigating user preferences and creating reliable recommendation systems.

2 Methodology

A recommendation system uses such data sets to filter information and predict user preferences or ratings for movies. Training on the extensive rating data, this project aims to build a model capable of analyzing user preferences, predicting ratings, and suggesting movies that align with user interests. Techniques such as collaborative filtering, content-based filtering, and hybrid models will be used to improve the precision of the recommendation and user satisfaction.

2.1 Demographic Filtering

A recommendation technique known as demographic filtering, rather than focusing on specific users, suggests well-liked products to all users based on their general popularity. The strategy is based on defining a scoring system that measures the quality of items in order to rank and suggest them.

The movie recommendation algorithm uses the IMDB weighted score as its metric. The following is the IMDB weighted rating formula $WR = \frac{v}{v+m}R + \frac{m}{v+m}C$ where v is the number of ratings for the movie; m is the minimum number of ratings, only movies with more than m ratings are recommended; R is the average rating of the movie; and C is average rating of all movies. The weighted ratings for every qualifying film can be generated for recommendation by computing the parameters C and m, and only films with more than m ratings are kept to guarantee the high standard of suggestions. The top N films can then be suggested to users by ordering the weighted ratings from highest to lowest.

2.2 Content Based Filtering

In order to make content-based recommendations, we first need to transform the content data into a quantifiable feature matrix. For example, features used in movie recommendation may include: Title as text type features; Genres as categorical features, which can be processed by labeling; and Director and Actors as discrete categorical features. After these features are preprocessed into textual data, they can be vectorized for representation by Feature Engineering. Commonly used methods include TF-IDF (Term Frequency-Inverse Document Frequency), which measures the importance of a word in a document while reducing the weight of high-frequency but low-distinguishing words.Therefore, the similarity between different items needs to be calculated. The most common method is Cosine Similarly, which has the following formula: Cosine Similarity $= \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|}$ where $\mathbf{A}$ and $\mathbf{B}$ are two eigenvectors, the numerator is the dot product of the vectors, and the denominator is the product of the modulus of the vectors. After calculating the similarity matrix of all the contents, input the feature vector corresponding to the target item (e.g., movie title), find the most similar item to it, and return the recommendation sorted by similarity.

2.3 Collaborative Filtering

Collaborative filtering is a recommendation method that uses user interaction data to recommend items that users may be interested in by examining similarities between users or items. In contrast to content-based recommendation, collaborative filtering bases its suggestions on user preferences or past behavior rather than the attributes of the items themselves.

The User-Based CF type is one. When the similarity between users is determined using cosine similarity or Pearson's correlation coefficient, items that are comparable to those that similar individuals favor are suggested for the target users. Based on the ratings of similar users, the target user's unrated items are estimated, assuming that similar users have similar rating preferences for similar goods. The other is Item-Based CF, which uses metrics like Pearson's correlation coefficient and cosine similarity to assess item similarity and recommends other items that are similar to the ones they've rated. Based on the target user's

evaluations of a certain item, forecast the ratings for more comparable things.

The introduction of latent factor models with singular value decomposition (SVD) addresses the scalability issue of collaborative filtering in high-dimensional sparse matrices. In this method, the implicit associations between users and items are captured by a dimensionality reduction technique, mapping them to a latent feature space. Lastly, reducing errors like RMSE and MAE helps maximize the model's prediction accuracy.

3 Results

3.1 Demographic Filtering

The *movies_metadata.csv* file used in this model contains the basic metadata for 45,000 films, such as title, average rating, popularity, *vote_count* (number of ratings), release date, etc. The two fields *vote_count* and *vote_average* are the focus of the model's implementation. $Vote\_count$ is the total number of ratings for each film, which is used to gauge its popularity based on the quantity of data; *vote_average* is the average rating for each film, which is a crucial sign of the film's quality. Some of the movies are substituted with a 0 for *vote_average* or a variable with an empty *vote_count* field in order to ensure that the rating computation is done correctly. Furthermore, resolve extreme anomalies in the number of ratings (*vote_count*) or ratings (*vote_average*) for select movies by filtering out movies with a rating count of zero.

Two crucial parameters are established in the recommendation model to guarantee the scientific character of the rating computation: the minimum number of ratings (m) and the average rating of all films (C). The average rating (C) for all films is determined to be 6.09, which means that most films have a concentration of medium ratings. The 90th quantile of the total number of ratings was selected as the threshold (m) to eliminate films with insufficient ratings; therefore, the number of ratings must be at least 1838 to be included in the list of suggested candidates. This decision prevents a worsening of recommendation quality brought on by an inadequate number of ratings and guarantees that the number of ratings of the suggested films is statistically significant. For instance, due to its high rating, a film with an 8.9 rating but only a 3 rating will not be given priority. By balancing the quantity and quality of ratings, this parameterization strengthens the recommendation list's credibility and dependability.

By filtering movies with a minimum threshold (m) number of ratings, we obtained 481 eligible movies. The films are then arranged from highest to lowest in order of their combined score (WR), which is determined using the IMDB weighted rating formula. The final output lists the top ten highest-rated films, which include classics like The Shawshank Redemption, Fight Club, and The Dark Knight. For example, The Shawshank Redemption has an average rating of 8.5 and a number of ratings of 8,205, which gives it an overall score of 8.06,

the highest of any movie. Even though Inception received a slightly lower average rating (8.1) than the other films, its composite score is still sixth due to its enormous number of ratings (13,752). The overall result demonstrates the effectiveness of the weighted rating formula in the recommendation system and the ability of the recommendation model to filter out high-quality films with a lot of ratings and high ratings in order to give users a solid foundation for recommendations.

3.2 Content Based Filtering

The plot summary of a film can be used as text data to calculate similarity in a simple way. The TF-IDF (Word Frequency-Inverse Document Frequency) method is used to convert the text data into numerical feature vectors first. This method lowers the weights of high-frequency meaningless words (such as "the,and,"etc.) and determines the importance of each word in the document. Cosine similarity is used to quantify the degree of similarity between films by computing the TF-IDF matrix for each film. Lastly, the user is presented with a list of the ten most comparable films based on the similarity score. The Dark Knight Rises was used as the TEST, and the top three recommendations were The Dark Knight, Batman Begins, and Batman Returns.The fundamental similarities between films can be effectively captured by this method, although it might only be applicable to films in the same genre or series. For example, the system does not identify other films by the same filmmaker, Christopher Nolan, that would be more relevant to the user's preferences when recommending The Dark Knight Rises.

To enhance the variety and quality of recommendations, more metadata about the film is added, including the director, main actor, genre, and keywords. Features like actor, director, and genre are first taken out of the raw data and transformed into strings that are normalized. The metadata of each film is then combined into an analyzable text data set, producing a "metadata soup"by splicing these features into a string. Moreover, the CountVectorizer is used to extract features from the data soup, and cosine similarity is used to determine the similarity. Again, using The Dark Knight Rises as a test, the top three recommendations after increasing the metadata are The Dark Knight, Batman Begins, and The Prestige (directed by Nolan). The technique can capture more movie correlations, including similarity in casting or directing style.

3.3 Collaborative Filtering

First, in user-based collaborative filtering, the Pearson correlation coefficient is utilized as a similarity measure, and it has been discovered that the similarity between user B and target user E is 0.87, indicating that their scoring preferences are comparable. In contrast, user D and target user E have a -1 similarity score, indicating that their preferences are completely different. Furthermore, cosine similarity is used to determine how similar two items are in item-based

collaborative filtering. The Matrix and Transformers have a similarity of 0.86, while The Avengers and Sherlock have a similarity of -1. Therefore, comparing the ratings of comparable people or similar goods can forecast the target user's unrated movies.

As an illustration of a more focused outcome, movies that user E has not rated are predicted using both item-based and user-based collaborative filtering. The Avengers movie received a user-based predictive rating of 3.51 and an item-based predictive rating of 3.62; the Sherlock movie received a user-based predictive rating of 3.81 and an item-based predictive rating of 3.87; the Me Before You movie received a user-based predictive score of 1.12 and an item-based predictive score of 1.15,etc. For both item-based and user-based collaborative filtering, the predicted scores are extremely similar, particularly for Sherlock and The Avengers. Compared to the other films, Me Before You had substantially lower scores, which is consistent with target user E's preference for less romantic themes.

The latent factor model (SVD) predicted user 1's rating of movie ID 302. The model predicted a rating of 2.58 with an error of -0.42, which was near to the actual number but slightly understated. The actual rating was 3.00. It is suggested that SVD's model is dependable in capturing user preferences and has good prediction accuracy since its error smaller than 0.5 and average RMSE of 0.89.

4 Conclusion

Demographic Filtering uses an IMDB weighted rating system to balance the quantity and quality of ratings, making it an easy and efficient way to suggest popular films. Fast recommendation generation without individualized user data is an a benefit, but there are serious drawbacks as well, like the inability to adapt suggestions according to user interests and the propensity to highlight popular, highly rated films.

Content-based filtering is a very effective and explanatory recommendation system that uses item attributes to make accurate recommendations. Even while plot synopsis-based suggestion works well in its basic form, the quality of recommendations is greatly enhanced by the addition of more detailed metadata. However, there are frequently no cross-genre recommendations and only possibilities that are extremely comparable to the supplied film. Adjusting feature weights or combining collaborative filtering algorithms can increase recommendation diversity and user happiness in the future.

Collaborative filtering yields individualized recommendation outcomes, however SVD outperforms in terms of prediction accuracy and applicability. In practice, different approaches, such as dynamic model update and hybrid recommendation can be combined adapt to users' changing preferences and improve recommendation outcomes. The SVD model's dependability in sparse data is further confirmed by example studies for user E's expected score and movie ID 302.