Web Intelligence

Project Report

Collaborative Filtering based Movie Recommendation Engine

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY

jiit

विद्या तत्व ज्योतिसमः

M.Tech CSE

Team Members:

1. Sidhant Moza (23303026)
2. Ehtesham Ashraf (23003021)
3. Himani Agrawal (23303025)

Table Of Contents

Abstract

In today's digital era, the vast amount of movie content available on platforms like IMDb poses a significant challenge for users in discovering new films tailored to their preferences. Collaborative filtering (CF) techniques have emerged as a promising solution for recommendation systems, leveraging user behavior data to generate personalized suggestions. This study focuses on User-Based Collaborative Filtering (UBCF), a subtype of CF, applied to IMDb's extensive movie database. By analyzing user ratings and similarities between users, UBCF identifies patterns and recommends movies that align with individual tastes. The methodology involves preprocessing raw data, computing user similarities using metrics such as cosine similarity, and generating recommendations based on nearest neighbors. Results demonstrate the efficacy of UBCF in providing accurate and personalized movie recommendations on IMDb, enhancing user experience and facilitating the exploration of diverse cinematic content. The findings underscore the potential of UBCF as a practical tool for improving movie discovery platforms, contributing to the advancement of recommendation systems in the entertainment industry

Collaborative Filtering Algorithm

- **Data Collection:** UBCF requires a dataset containing information about user interactions with items, such as purchases, ratings, views, likes, and reviews. This data is typically collected through user activity tracking systems implemented on e-commerce websites or online platforms.

- **Similarity Computation:** The first step in UBCF involves computing the similarity between items based on the historical behavior of users. Various similarity metrics can be used, such as cosine similarity, Pearson correlation coefficient, or Jaccard similarity. These metrics quantify the degree of similarity between pairs of items based on user interactions.

$$sim(i,j) = \frac{\langle R_{k,i} - A_i,\ \ R_{k,j} - A_j \rangle}{\|R_{k,i} - A_i\| \| R_{k,j} - A_j\|} = \frac{\sum_{k=1}^{n}(R_{k,i} - A_i)\ (R_{k,j} - A_j)}{\sqrt{\sum_{k=1}^{n}(R_{k,i} - A_i)^2 \times \sum_{k=1}^{n}(R_{k,j} - A_j)^2}}$$

- **Neighborhood Selection:** Once the similarity between items is calculated, a neighborhood of similar items is selected for each item in the dataset. This neighborhood typically consists of the top N most similar items to a given item. The size of the neighborhood (N) can be predefined or determined dynamically based on the dataset and performance requirements.

- **Prediction Generation:** After selecting the neighborhood of similar items for each item in the dataset, predictions are generated for items that a user has not interacted with. This is achieved by aggregating the preferences or ratings of similar items weighted by their similarity to the target item. Common aggregation methods include weighted averages, weighted sums, or regression techniques.

$$P_{u,i} = A_u + \frac{\sum_{w=1}^{n}\left(R_{w,i} - A_w\right) \times sim(u,w)}{\sum_{w=1}^{n}\left|sim(u,w)\right|}$$

- **Recommendation Generation:** Finally, the algorithm generates a list of top recommendations for each user based on the predictions generated in the previous step. These recommendations are ranked based on the predicted preferences or ratings, and the top-ranked items are presented to the user as personalized recommendations.

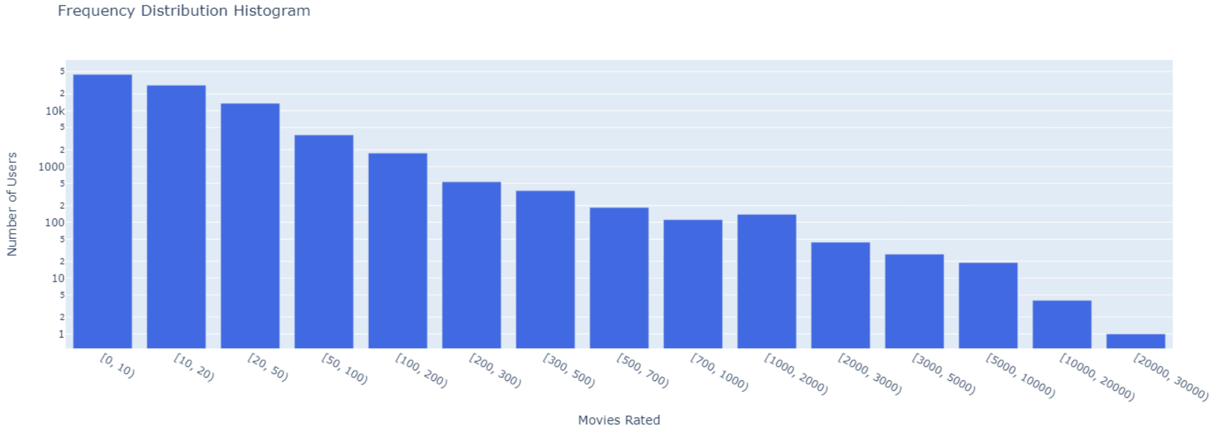
UBCF works by leveraging the collective wisdom of users to make personalized recommendations. Instead of relying solely on the preferences of individual users, UBCF identifies patterns and similarities in user behavior to recommend items that are likely to be of interest to a given user. By analyzing historical interactions between users and items, UBCF identifies similar items and uses this information to make predictions about user preferences for new items.

Methodology

Exploratory Data Analysis:

The project begins with exploratory data analysis on the IMDb ratings dataset obtained from Kaggle. The dataset consists of user IDs, title IDs, ratings, and rating dates. The data is loaded into a pandas DataFrame, and duplicate entries are removed. The rating dates are converted to datetime format, and new columns are created to extract day, month, year, and day of the week information from the date.

The distribution of ratings across days, months, years, and days of the week is visualized using count plots. The top 10 most-rated movies and users with the highest number of ratings are identified. The distribution of the number of ratings per user is also analyzed and plotted.



Collaborative Filtering Implementation:

The collaborative filtering algorithm is implemented from scratch without using any external libraries. The first step is to select a sample of 10 users who have rated at least 5 of the top 10 most-rated movies. A subset of the dataset is created containing only the ratings from these 10 users for the top 10 movies.

User-based Collaborative Filtering:

User-based collaborative filtering is performed on the sample dataset. The user-item rating matrix is constructed, and the cosine similarity between each pair of users is calculated. The ratings are mean-centered before computing the similarity.

Rating Prediction:

Two modes of rating prediction are demonstrated:

1. Predicting the rating a specific user would give to a specific movie.
2. Predicting the top N movies for each user, along with the predicted ratings.

The prediction is made by taking the weighted average of the mean-centered ratings from similar users, where the weights are the similarities between the target user and the other users.

Evaluation Metrics:

The predicted ratings are evaluated using the following metrics:

1. Mean Absolute Error (MAE)

   $$MAE \ = \ \frac{1}{N}\sum_{i}\left|y_t - y_p\right|$$

2. Mean Squared Error (MSE)

   $$MSE \ = \ \frac{1}{N}\sum_{i}(y_t - y_p)^2$$

3. Root Mean Squared Error (RMSE)

   $$RMSE \ = \ \sqrt{MSE}$$

4. Normalized Discounted Cumulative Gain (NDCG)

   $$NDCG \ = \ \frac{DCG_k}{IDCG_k}$$

   Where, $DCG_k \ = \ \sum_{i=1}^{k} \frac{Rating_i}{log_2(i)}$

   and IDCG is the DCG of an ideal rating sequence i.e, in descending order.

The NDCG score measures the quality of the ranking of the predicted ratings.

Collaborative Filtering on the Entire Dataset:

A random user with more than 500 ratings is selected from the dataset. The user's ratings are split into training and test sets, with 20% of the ratings used for testing. Similar users are identified based on the cosine similarity of their ratings on the common movies in the training set.

The rating predictions for the test set movies are made using the weighted average of the mean-centered ratings from similar users. The predictions are evaluated using MAE, MSE, RMSE, and NDCG.

Performance Evaluation:

The collaborative filtering engine is evaluated on all users who have rated at least 500 movies. For each user, the rating prediction and evaluation process is repeated. The average MAE, MSE, RMSE, and NDCG scores across all users are computed to assess the overall performance of the collaborative filtering engine.

Results

Exploratory Data Analysis:

- The dataset contains 4,669,820 ratings from 1,499,238 users for 351,109 movies.
- The top 10 most-rated movies are identified, including titles like "The Shawshank Redemption," "The Dark Knight," and "Avengers: Endgame."
- The distribution of the number of ratings per user is heavily skewed, with most users having rated fewer than 50 movies.

Collaborative Filtering on Sample of Users:

- User-based collaborative filtering is successful in predicting ratings for the sample dataset of 10 users and the top 10 movies.
- The predicted ratings and top N recommendations for each user in the sample are displayed.

Collaborative Filtering on the Entire Dataset:

For the randomly selected user with 601 ratings, the collaborative filtering engine achieves:

- Mean Absolute Error (MAE) = 0.825
- Mean Squared Error (MSE) = 1.003
- Root Mean Squared Error (RMSE) = 1.001
- Normalized Discounted Cumulative Gain (NDCG) = 0.889

Performance Evaluation on 530 Test Users:

The collaborative filtering engine is evaluated on 530 users who have rated more than 500 movies.

The average performance metrics across these users are:

- Average MAE = 1.356
- Average MSE = 3.208
- Average RMSE = 1.727
- Average NDCG = 0.816

Discussion

The collaborative filtering engine implemented from scratch demonstrates promising results in predicting movie ratings and generating personalized recommendations. The exploratory data analysis reveals the skewed distribution of the number of ratings per user, which is a common characteristic of rating datasets.

The user-based collaborative filtering approach effectively leverages the similarities between users' rating patterns to make predictions. The cosine similarity metric is used to measure the similarity between users based on their mean-centered ratings, which helps to account for individual rating biases.

The evaluation metrics, including MAE, MSE, RMSE, and NDCG, provide insights into the accuracy and ranking quality of the predictions. The NDCG score, which takes into account the ranking of the predicted ratings, is particularly useful for evaluating the quality of the top recommendations.

While the performance of the collaborative filtering engine is promising, there is still room for improvement. The average RMSE of 1.727 indicates that the predictions may deviate from the true ratings by approximately 1.7 points on the rating scale. Additionally, the average NDCG of 0.816 suggests that the ranking of the top recommendations could be further optimized.

Several factors may contribute to the observed performance:

1. **Data sparsity:** The dataset is likely to be sparse, with many users having rated only a small subset of movies. This can make it challenging to find reliable similarities between users, affecting the accuracy of predictions.
2. **Cold-start problem:** The collaborative filtering engine may struggle to make accurate predictions for new users or movies with few ratings, as there is limited information to establish similarities or patterns.
3. **Temporal dynamics:** The dataset may not capture changes in user preferences over time, which could lead to less accurate predictions for more recent or older movies.
4. **Implicit feedback:** The current implementation only considers explicit ratings, but incorporating implicit feedback signals, such as movie viewing history or browsing behavior, could potentially improve the recommendations.

Conclusion

A collaborative filtering-based movie recommendation engine was successfully implemented from scratch, demonstrating its effectiveness in predicting ratings and generating personalized recommendations. The exploratory data analysis provides valuable insights into the dataset, and the step-by-step implementation of the collaborative filtering algorithm is presented in detail.

The evaluation results, including the average MAE, MSE, RMSE, and NDCG scores, indicate the overall performance of the engine and highlight areas for potential improvement. While the performance is promising, there is room for further optimization and enhancement of the collaborative filtering algorithm.

Future work could explore incorporating additional features or techniques to address challenges such as data sparsity, the cold-start problem, and temporal dynamics. Hybrid approaches that combine collaborative filtering with content-based or knowledge-based methods could also be explored to enhance the recommendation quality.