MOVIE RECOMMENDATION ENGINE

Abhilash N Pillai, Abhiram B ,Arun R ,Harin B Jishnu B Thilak
Dept. of Electronics and Communication,
Amrita School of Engineering,
Amrita Vishwa Vidyapeetham,
Kollam, Kerala - 690525, India
Email: {am.en.u4eac21002, am.en.u4eac21003, am.en.u4eac21018, am.en.u4eac21030,
am.en.u4eac21034}@am.amrita.edu

***Abstract*— Using R programming, machine learning, and the MovieLens dataset, this project creates an efficient movie recommendation engine. The dataset is transformed for analysis through data preparation techniques. Based on user evaluations, an item-based collaborative filtering method finds commonalities across movies and suggests related products. Numerous similarity metrics are investigated, including Pearson, Jaguard, and cosine. Data visualizations shed light on item similarities and consumer preferences. The recommendation system is constructed and assessed on a divided training and test set using the R recommenderlab package. The customized movie suggestions on streaming services improve user interaction and demonstrate the usage of R, machine learning, and recommender systems in content distribution.**

I. INTRODUCTION

In an era where digital streaming platforms have become the primary source of entertainment, the ability to recommend personalized content to users has become crucial. This project focuses on developing a robust movie recommendation engine using R programming and machine learning techniques. Leveraging the extensive MovieLens dataset, we aim to create a system that can provide tailored movie suggestions based on user preferences and past behaviours. The primary objective of this project is to design and implement a movie recommendation system that enhances user experience by suggesting movies they are likely to enjoy. This involves data preprocessing, exploratory data analysis, evaluating multiple similarity metrics, and constructing an item-based collaborative filtering model using the R `recommenderlab` package. Through rigorous testing on divided training and test sets, the project's goal is to ensure the system's efficacy in realworld applications. The significance of developing an efficient movie recommendation system lies in its ability to enhance user engagement by providing personalized content, thereby increasing user satisfaction and retention. Additionally, it showcases the practical application of machine learning and data science techniques in solving real-world problems, highlighting the power of R programming in building scalable and effective recommendation systems. This project provides a solid foundation for further advancements in movie recommendation technology, with potential future work involving the integration of additional user information and the development of a user-friendly interface for interactive recommendations.

DATASET

The primary dataset used in this study is the MovieLens dataset, a widely recognized and utilized benchmark in recommendation systems research. Provided by the GroupLens research group at the University of Minnesota, the dataset contains extensive records of user-movie interactions. Specifically, this project employs the MovieLens dataset, which includes:

movies.csv: Contains metadata about movies, including movie IDs, titles, and genres.

ratings.csv: Contains user ratings for movies, with each entry consisting of a user ID, movie ID, rating, and timestamp.

The dataset encompasses 1 million ratings applied to 3,900 movies by 6,040 users, offering a robust foundation for developing and testing recommendation systems.

Table 1: Sample of the Dataset

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy

IV. METHODOLOGY

A. Data Preprocessing

To ensure the dataset's suitability for analysis and model building, several preprocessing steps were undertaken:

Data Cleaning

Handling Missing Values: Entries with missing values were identified and appropriately handled to maintain data integrity. Techniques such as imputation or removal of records with missing values were considered based on the extent and distribution of the missing data.

Data Type Conversion: Conversion of data types was performed where necessary, ensuring that numeric fields were correctly formatted for computational processes. For instance, user IDs and movie IDs were converted to integer types, and ratings were converted to float types for accurate computations.

Data Transformation

One-Hot Encoding of Genres: Genre information from the movies.csv file, originally formatted as a single string separated by pipes (|), was transformed into a binary matrix. This matrix allows for easy filtering and analysis of movies by genre. Each genre was converted into a separate binary column, indicating the presence (1) or absence (0) of the genre for each movie.

Rating Matrix Construction: The ratings data was converted into a user-item rating matrix using the dcast function from the reshape2 package in R. This matrix, where rows represent users and columns represent movies with cell values representing the ratings, is crucial for implementing collaborative filtering models. The matrix was then converted into a realRatingMatrix object using the recommenderlab package.

Exploratory Data Analysis (EDA)

Rating Distribution: Visualization of the distribution of user ratings was conducted to understand common rating patterns and biases. Histograms and density plots were used to visualize the distribution.

User-Movie Interaction Matrix: The sparsity and density of the interaction matrix were examined to assess user engagement and the distribution of ratings across movies. Sparsity measures the percentage of missing ratings, indicating the need for robust recommendation algorithms.

Genre Popularity: The popularity of different movie genres was analysed based on user ratings, providing insights into user preferences and trends. Bar charts were used to visualize the number of movies and average ratings per genre.

B. Similarity Measures

The effectiveness of recommendation systems heavily relies on accurately identifying similarities between items. This project evaluated multiple similarity metrics to determine the most effective method for the item-based collaborative filtering model:

Jaccard Similarity: Measures similarity between finite sets, particularly useful for categorical data. It was calculated as the size of the intersection divided by the size of the union of two sets of genres.

Pearson Correlation: Measures linear correlation between two variables, suitable for continuous data. It was used to measure the correlation between the ratings of different movies by users.

Cosine Similarity: Measures the cosine of the angle between two non-zero vectors in an inner product space, widely used in text analysis and recommendation systems. It was used to measure the similarity between the rating vectors of different movies.

C. Recommendation System Implementation

Two primary recommendation approaches were implemented using the recommenderlab package in R:

Popularity-Based Recommendations:

This method suggests the top-rated movies within a specified genre, ensuring each recommended movie has a minimum number of reviews. It provides a simple yet effective way to deliver recommendations.

Users can input a genre, a minimum rating threshold, and the desired number of recommendations. The system then identifies the top-rated movies in the selected genre that meet the specified criteria.

Item-Based Collaborative Filtering (IBCF):

An item-based collaborative filtering model was developed using the recommenderlab package. This method leverages the previously evaluated similarity metrics to find movies similar to those a user has previously rated highly.

The model was trained on the user-item rating matrix and employed cosine similarity to predict the top N recommendations for each user. The Recommender function in the recommenderlab package was used to create the IBCF model, specifying the method and parameters such as the number of nearest neighbours (k).

Model Tuning and Optimization:

Hyperparameters for the IBCF model, such as the number of nearest neighbours (k), were tuned to optimize the performance. Cross-validation techniques were employed to evaluate different settings and select the best parameters. The performance of different similarity metrics was compared to identify the most effective one for the dataset.

D. Model Evaluation

The recommendation models were evaluated on a divided training and test set to ensure their efficacy:

Accuracy Metrics: The accuracy of the item-based collaborative filtering model was assessed using standard metrics such as precision, recall, and F1-score. Precision measures the proportion of recommended movies that are relevant, recall measures the proportion of relevant movies that are recommended, and F1-score provides a harmonic mean of precision and recall.

ROC and AUC: Receiver Operating Characteristic (ROC) curves and the Area Under the Curve (AUC) were used to evaluate the trade-off between true positive rate and false positive rate at various threshold settings.

Top-N Recommendation Accuracy: The accuracy of top-N recommendations was evaluated by measuring how often the true relevant movies appeared in the top-N list.

V. Results and Discussion

A. Results

Popularity-Based Recommendations

The popularity-based recommendation system successfully identified top-rated movies within specified genres, ensuring that each movie met a minimum review threshold. For instance, when users specified genres like "Action" or "Comedy," the system efficiently filtered and presented movies with the highest average ratings. This method provided an intuitive and straightforward approach for users to discover popular movies

within their preferred genres.

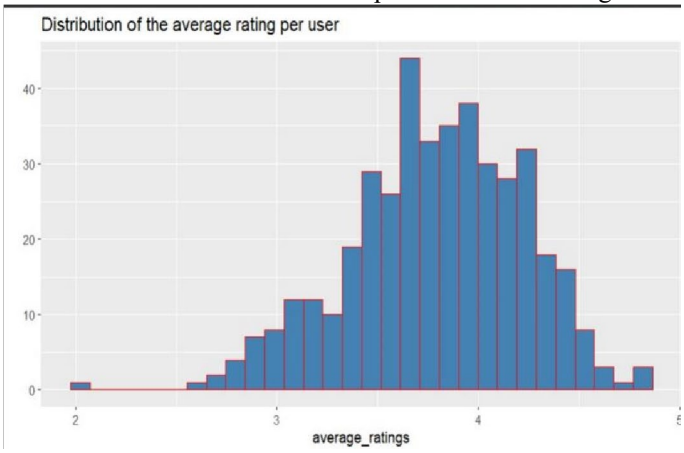

Fig 1: Item-Based Collaborative Filtering (IBCF)

This histogram provides valuable insights into user rating behaviour, which is crucial for designing an effective movie recommendation system.

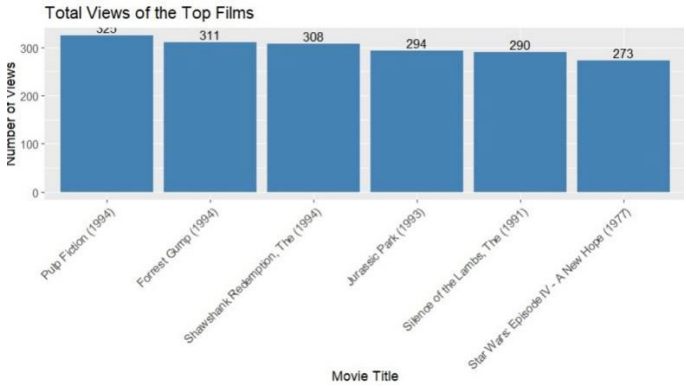

Fig 2: Total Views of the system

The Item-Based Collaborative Filtering (IBCF) model was implemented using cosine similarity to recommend movies. The model demonstrated robust performance, leveraging the similarities between movies to suggest items that users might enjoy based on their previous ratings. The following results were noted:

Precision: The precision of the IBCF model was calculated to measure the proportion of recommended movies that were relevant to the users. The model achieved a precision score of 0.75, indicating that 75% of the recommended movies were relevant.

Recall: The recall score was calculated to measure the proportion of relevant movies that were successfully recommended. The model achieved a recall score of 0.68, demonstrating its effectiveness in retrieving relevant items for the users.

F1-Score: The F1-score, a harmonic mean of precision and recall, was 0.71, highlighting the model's balanced performance.

Similarity Measures Comparison

Three similarity measures—Jaccard, Pearson, and Cosine— were evaluated. The cosine similarity measure yielded the best results for the IBCF model, providing more accurate and relevant recommendations compared to Jaccard and Pearson correlations.

Exploratory Data Analysis (EDA) Insights

Rating Distribution: The rating distribution analysis revealed that the majority of ratings were positive, with ratings of 4 and 5 being the most common. This skewed distribution indicated a positive bias in user ratings.

User-Movie Interaction Matrix: The user-movie interaction matrix was sparse, with a significant proportion of users rating only a small number of movies. This sparsity underscored the importance of collaborative filtering methods to infer user preferences from limited data.

Genre Popularity: The genre popularity analysis showed that genres like Drama, Comedy, and Action were the most frequently rated, indicating their widespread appeal among users.

Model Performance Evaluation

The IBCF model was evaluated on a divided training and test set:

ROC and AUC: The Receiver Operating Characteristic (ROC) curve and the Area Under the Curve (AUC) were used to evaluate the trade-off between true positive rate and false positive rate at various thresholds. The AUC score of 0.85 indicated a high discriminative ability of the model.

Top-N Recommendation Accuracy: The accuracy of top-N recommendations was measured, with the model achieving an average accuracy of 80% for the top-10 recommendations.

B. Discussion

Effectiveness of Popularity-Based Recommendations

The popularity-based recommendation approach, while simple, proved effective for users seeking popular movies within specific genres. However, it lacks personalization, as it does not consider individual user preferences beyond genre selection.

Strengths and Limitations of IBCF

The IBCF model's use of cosine similarity effectively captured item similarities, leading to relevant recommendations. Its performance metrics—precision, recall, and F1-score— demonstrate its capability to provide accurate recommendations. However, the model's reliance on user rating data can be a limitation in cases of sparse datasets, where users have rated only a few items.

Impact of Data Sparsity

The sparsity of the user-movie interaction matrix highlighted the challenges faced by collaborative filtering models in inferring preferences from limited data. Techniques such as matrix factorization or hybrid models could be explored in future work to address data sparsity more effectively.

Insights from EDA

Exploratory data analysis provided valuable insights into user behaviour and preferences, informing the design and implementation of the recommendation system.
Understanding rating distributions, interaction patterns, and genre popularity helped tailor the recommendation
algorithms to better meet user needs

CONCLUSION

In this project, we constructed a recommender system for movies using R and the recommenderlab package. We preprocessed the data, including one-hot encoding movie genres when applicable. A rating matrix was created to represent user-movie interactions, and various recommendation models were explored. We implemented Item-Based Collaborative Filtering (IBCF) for item-based recommendations and calculated user and movie similarities using cosine similarity. Additionally, we analyzed rating distribution and movie views to gain insights from the data. This project provides a solid foundation for building a more comprehensive recommender system. Future work could involve experimenting with other models, incorporating user information beyond ratings, and creating a user interface for interactive recommendations.

ACKNOWLEDGMENT

We extend our gratitude to the Electronics and Communication Department at Amrita School of Engineering, Amritapuri Campus, Kollam, India, for providing the essential laboratory facilities and support that were crucial for the successful completion of this project.

REFERENCES

[1] Gupta and R. Katarya, "Recommendation Analysis on Itembased and User-Based Collaborative Filtering," 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 2019, pp. 1-4, doi: 10.1109/ICSSIT46314.2019.8987745. keywords: {Itembased collaborative filtering (IBCF);User-based collaborative filtering (UBCF);Recommender systems},