User Behavior Analysis for Improved Movie Recommendations

P32 - Niharika Maruvanahalli Suresh Surabhi Ravindran Nair Siri Paidipalli

1 Background and Introduction

In today's world, content consumption has reached exceptional levels and is driven by changing consumer behaviors. This necessitates the need for a personalized experience. The rapid growth of data has led to advanced data collection techniques such as data mining, web scraping, etc. to ensure that the user is presented with relevant and high-quality content. This new era of information creates more efficient systems that can promise user satisfaction and retention. However, there is abundant information in these platforms, making it challenging for the end user to find the one that aligns with their tastes. Here's where an efficient recommendation comes into play by assisting the user with tailored experiences thus simplifying the content discovery and enhancing the overall engagement.

1.1 Recommendation systems

A recommendation system is a type of information filtering system designed to predict and suggest items or content that a user may find relevant or interesting. These systems collect data and then analyze and decode the pattern in the user's behavior. This analysis is then used to determine which information is more relevant to the user and ultimately improve the user's quality of search results and personalized recommendations. This helps users navigate through large volumes of data more resourceful. In recommendation systems, three common types of filtering methods are used to make suggestions relevant to the user based on different principles of user interaction.

- **Demographic filtering** – This method provides recommendations based on the popularity of the movie, genre, and demographic data thus focusing on broader trends. However, this provides more generalized recommendations to every user and assumes that users from the same demographic area will have similar preferences. The basic idea behind this system is that movies that are more popular and critically acclaimed will have a higher probability of being liked by the average audience.

- **Content-based filtering** - This method suggests products to the user based on the previous choices that the user has made, i.e. this system highly depends on the user's metadata and analyzes the user's characteristics rather than the user's behavior to provide recommendations. The general idea behind the recommendation systems is that if a person likes a particular item, he or she will also like an item that is similar to it.

- **Collaborative filtering** – This system matches users with similar interests and preferences and provides recommendations. Collaborative filtering follows the principle that like-minded people will have similar tastes and interests and therefore recommend items based on the user's behavior. Thus, this method does not require metadata of the user's previous activities.

1.2 Streaming platforms which use recommendation systems

Many streaming platforms use recommendation systems to improve the quality of the suggestions and boost user satisfaction. Some of them are as follows: **Amazon** uses a recommendation system to suggest products based on a user's browsing history, past purchases, and items viewed by similar

users which act as a driving force for the sales and the shopping experience. **YouTube** uses a recommendation system to decide what to play next and what kind of videos to show on the homepage based on the user's watch history, likes, and behavior of users with similar preferences, thus making sure the content keeps the user engaged on the platform. **Facebook** uses a recommendation system to suggest pages, groups, and friends to users based on their activity and behavior of their network which helps in directing them to new relevant pages and new friends to connect with. Similarly, entertainment sites such as **Netflix and Spotify** employ advanced recommendation systems to suggest movies TV shows or music that align with their preferences which contribute significantly to user retention and overall platform success.

1.3 Rating a product

User engagement has become a crucial factor while designing a system because it directly impacts the likelihood of continued subscription, platform loyalty, more content consumption, and increased watch time. Also, higher engagement provides more customer interaction data which can be used to further improve the system. Accuracy of the recommendations, the design interface of the platform, the variety in the content, content based on real-time trends, and adaptive algorithms are some factors that promise high user engagement.

Another essential element in the data feedback loop that allows the system to better understand individual tastes is Rating a product. When a user positively or negatively rates a movie, it provides direct insight into their likes or dislikes. Ratings help Netflix differentiate between content the user simply tolerates versus content they truly love. This feedback enables the system to adapt to changing preferences, identify outliers, reduce guesswork, and refine the recommendation model. Detecting the pattern of users with similar likes/ dislikes increases the chances of prioritizing content relevant to the user. Users' inputs matter as they build trust in the recommendation system, leading to stronger user-system relationships.

2 Proposed Method

In this project, we aim to build a personalized movie recommendation system using a hybrid approach that integrates Collaborative Filtering and Matrix Factorization techniques, specifically Singular Value Decomposition (SVD). The following sections provide the intuition and describe the algorithms.

2.1 Intuition - why should it be better than the state of the art?

2.1.1 Handling Sparsity

Sparse datasets are a common challenge for collaborative filtering. By employing dimensionality reduction techniques like Singular Value Decomposition (SVD), the system can handle the sparsity of user-movie interactions effectively, capturing latent relationships between users and items

2.1.2 Improved Scalability

By incorporating a sparse matrix representation and computationally efficient algorithms like k-Nearest Neighbors (k-NN) and SVD, the system can scale to large datasets like the Netflix Prize dataset, by lowering the computational complexity of similarity calculations and recommendations.

2.1.3 Enhanced Context Awareness and Personalization

User-based filtering ensures highly tailored recommendations by recognizing people with similar preferences. Item-based filtering assesses movie similarities, resulting in contextually relevant recommendations. By combining these techniques, a more comprehensive recommendation experience is produced that takes into account both item-level similarities and user-specific preferences.

2.1.4 Cold Start Issue Mitigation

Our technique uses item similarity and SVD's latent features to deliver superior recommendations, even for lower-rated films or new users, whereas standard collaborative filtering frequently fails for new users or things with minimal data.

2.1.5 Error Reduction

Errors are decreased by combining user-, item-, and SVD-based predictions. The accuracy of our forecasts is confirmed by metrics such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), which show gains over stand-alone methods.

2.2 Description of Algorithms

2.2.1 Data Pre-processing

We have used the Netflix Prize Dataset, which contains over 100 million of user ratings for various movies. This dataset is particularly well suited for collaborative filtering and matrix factorization due to its large scale and variety of user-movie interactions.

The data pre-processing phase is crucial for preparing the dataset in a structured manner, ensuring that the data is clean and ready for modeling. This particular dataset consists of two primary sources – movie catalogue and customer review data. The movie catalogue includes details such as movieID , release year and movie titles which are loaded from a csv file, which are saved in Data Frame; where the movie ID is set as the index to enable easier lookup during the subsequent stages. For customer reviews, the data is loaded from multiple files and ratings are standardized to float format for consistency across datasets. Missing values are handled by identifying and removing NaN values from the ratings, ensuring no empty entries are processed. A new MovieId column is added by mapping ratings to corresponding movies. Both CustId and MovieId are converted to integers to ensure data uniformity. This cleaned dataset is ready for efficient use in the recommendation model.

In order to incorporate collaborative filtering, the matrix of user-item interactions is created. This matrix is built by selecting the top 1000 users and top 1000 movies with the most ratings, emphasizing high-quality, frequent interactions in the dataset. A pivot operation is executed to reformat the data into a matrix structure with users as rows, movies as columns, and ratings as values. The resulting matrix is converted into a sparse matrix format, which decreases memory consumption and enhances performance, particularly with large datasets like the Netflix prize data. This processed sparse matrix serves as the basis for the following recommendation modelling stages.

2.2.2 Collaborative Filtering

Collaborative filtering is the first phase of our recommendation system. This technique aims to predict user's interests by collecting preferences from similar users or items.The recommendation is designed to implement both user-based and item-based collaborative techniques, combined with the K-nearest neighbors algorithm. This aims at identifying either users with similar movie-watching behavior or items that are similar to those the user has already watched. The system will then recommend movies based on these similarities.

The KNN model is built using the cosine similarity, which measures the similarity between users based on their movie rating patterns. This makes it possible for us to identify users who have given comparable ratings to films. The model is trained on a spare user-item interaction matrix, where rows represent users, and columns represent movies. Once the model is fit, we can identify the neighbor(similar user) for any given user. For user-based collaborative filtering, the system recommends movies based on the average ratings from similar users, excluding movies the target user has already seen. The code also implements item-based collaborative filtering by compiling cosine similarity between movies. The system looks for the similarity between movies a user has rated and other movies in the dataset and it recommends a movie that is similar to those the user has already rated. This method is helpful when a user's profile is sparse but there's a lot of information about movies themselves.

We have combined both user based and item based collaborative filtering which provides the robust recommendations. We have tested the following collaborative filtering models:

- Collaborative filtering based on users: Suggestions for a user are created using the evaluations given by their closest neighboring users.
- Item-based collaborative filtering: Suggestions are given by looking at movies that are comparable to ones the user has already rated positively.

Both methods offer different advantages that complement each other. By leveraging them together, we aim to capture both user preferences and item similarities, ensuring a richer recommendation experience.

2.2.3 Matrix Factorization with SVD

The proposed algorithm leverages **Singular Value Decomposition (SVD)** to generate personalized movie recommendations by extracting latent features from a user-item interaction matrix. This interaction matrix represents users as rows, movies as columns, and the corresponding ratings as matrix values. The SVD decomposes the sparse interaction matrix into three key components:

1. **User Feature Matrix (U)**: Encodes user-specific latent features.
2. **Singular Values (Σ)**: Represents the importance of the latent factors in descending order.
3. **Movie Feature Matrix (V^T)**: Encodes latent features of the movies.

This decomposition identifies underlying patterns that capture user preferences and movie attributes, even in sparse datasets. To reduce computational complexity and focus on critical information, only the top k singular values (e.g., $k = 200$) are retained. This dimensionality reduction compresses the data while preserving key latent features. Using the reduced matrices, a lower-rank approximation of the original interaction matrix is reconstructed. The predicted ratings matrix is calculated as:

$$PredictedRatingsMatrix = U_k \cdot \Sigma_k \cdot V_k^T \tag{1}$$

where U_k, Σ_k, and V_k^T are the truncated forms of U, Σ, and V^T, respectively. Each entry in the predicted matrix represents an estimated rating that a user would assign to a movie they have not yet rated.

Recommendation Generation A function, `recommend_movies_svd`, is implemented to generate personalized recommendations for a specific user. The procedure is as follows:

1. **Retrieve Predicted Ratings:** Access the row of the predicted ratings matrix corresponding to the selected user.
2. **Exclude Rated Movies:** Remove movies that the user has already rated to avoid redundancy.
3. **Sort and Select:** Sort the remaining movies by their predicted ratings in descending order and select the top N movies for recommendation.

By capturing latent user preferences and hidden movie attributes, the SVD-based algorithm effectively personalizes recommendations for individual users. This approach demonstrates the robustness and scalability of matrix factorization techniques in modern recommendation systems.

2.2.4 Evaluation Matrix

The project implements a movie recommendation system using Singular Value Decomposition (SVD) and collaborative filtering techniques, with an emphasis on robust evaluation. To assess the system's performance, the dataset is divided into training and test sets through a **test-train split. A proportion of user ratings is randomly selected for the test set, while the corresponding entries in the training set are set to zero, ensuring these ratings are "hidden" during model training. The training set is then used to build the model, and the predicted ratings for the test set are compared with the actual ratings to evaluate the model's accuracy.

SVD is applied to the training data to extract latent factors, reducing the dimensionality of the user-item matrix and enabling predictions for unrated items. Collaborative filtering further enhances the system by employing user-based filtering, which identifies similar users to predict ratings, and item-based filtering, which recommends movies based on item similarities. To validate the system, evaluation metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) are computed. RMSE highlights the model's sensitivity to larger errors, while MAE provides an intuitive measure of the average prediction error. By combining these metrics with a rigorous test-train split methodology, the system ensures accurate, scalable, and reliable recommendations for real-world applications.

3 Plan & Experiment

This study explores the effectiveness of various recommendation techniques using the **Netflix Prize Dataset**, which contains over 100 million ratings from approximately 480,000 users for 17,000 movies. The dataset's sparsity—where most users rate only a small subset of movies—poses a significant challenge for traditional collaborative filtering methods, making it an ideal benchmark for testing diverse algorithms.

The aim is to evaluate and compare three models: user-based collaborative filtering, item-based collaborative filtering, and matrix factorization using Singular Value Decomposition (SVD). Furthermore, the study examines whether a hybrid model that combines item-based filtering and SVD can outperform standalone methods. The ultimate objective is to develop a recommendation system capable of providing accurate and personalized suggestions while addressing sparsity issues.

3.1 Objectives

- Assess whether item-based collaborative filtering is more effective than user-based methods due to its ability to leverage stable item relationships in sparse datasets.
- Evaluate the performance of matrix factorization via SVD, which identifies hidden features to capture complex user-item interactions.
- Investigate the potential of a hybrid approach that integrates item-based filtering with SVD to enhance recommendation accuracy and relevance.

3.2 Experimental Setup

The dataset underwent preprocessing, including rating normalization and a split into **80% training** and **20% testing** sets. This ensured ample data for training while reserving unseen samples for reliable evaluation of the models.

Initially, user-based and item-based collaborative filtering techniques were applied using K-Nearest Neighbors (KNN) with cosine similarity to recommend movies. The performance was evaluated using Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). Due to dataset sparsity, the user-based method yielded high errors (RMSE = 4.0608, MAE = 4.06), highlighting its limitations in sparse datasets. Conversely, the item-based method performed significantly better with RMSE = 0.7710 and MAE = 0.7709, demonstrating its ability to handle sparse data effectively by focusing on similarities between items rather than users. The system recommends movies based on the preferences of similar users and also identifies movies similar to those already rated by user . The graphs below shows the top 5 recommendations for User 305344, along with their predicted ratings and item based recommendations.

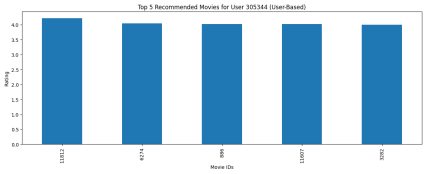

Figure 1: User based recommendations

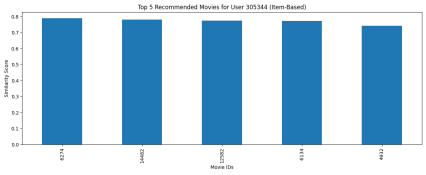

Figure 2: Item based recommendations

To improve accuracy, matrix factorization with Singular Value Decomposition (SVD) was implemented to uncover latent features underlying user-item interactions. This method outperformed KNN-based approaches, achieving RMSE = 0.7506 and MAE = 0.5932, confirming its robustness in sparse scenarios. By factoring the user-item interaction matrix into smaller latent matrices, SVD provided a more nuanced understanding of user preferences, enhancing the precision of recommendations. The improved accuracy underscored SVD's effectiveness in addressing challenges posed by sparsity in large datasets. By factorizing the user-item interaction matrix into latent features, SVD predicts ratings for unrated movies. The graph below highlights the top 5 SVD-based recommendations for User 305344.

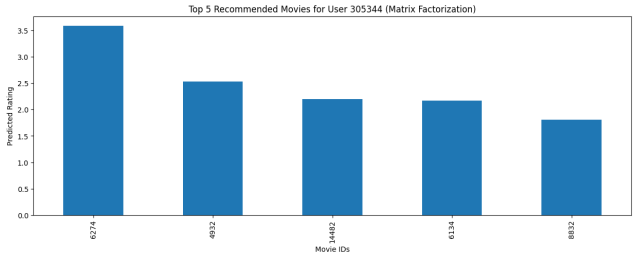

Figure 3: Matrix Factorization recommendations

Finally, a hybrid model was developed, combining predictions from item-based collaborative filtering and SVD. This integrated approach leveraged the strengths of both methods, delivering the lowest errors and achieving the best overall performance. By synthesizing the complementary insights of collaborative filtering and matrix factorization, the hybrid model not only reduced prediction errors but also enhanced the relevance of the recommendations. These findings illustrate the value of combining diverse techniques to overcome the limitations of individual models, ultimately improving the quality and accuracy of recommender systems.

4 Results

We performed a comprehensive analysis of the experimental results obtained using the 3 approaches above using metrics such as Root Mean Squared Error(RMSE), Mean Absolute Error(MAE), Precision and Recall to judge their performance and reliability.

4.1 Quantitative Results

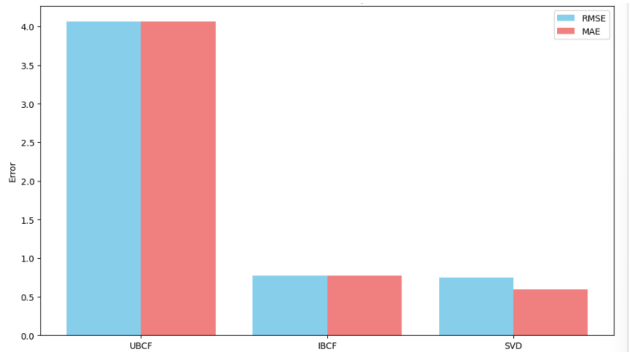

Figure 4: Error Rates

4.1.1 Error Rate Analysis

We assessed RMSE and MAE, both measure the difference between predicted and actual values. RMSE is sensitive to large errors because it squares the differences before averaging them. Unlike

RMSE, MAE does not square the errors, making it less affected by the outliers. The results are as follows:

1. **User-based Collaborative Filtering (RMSE = 4.0608, MAE = 4.06)**
   The error rates are considerably high which indicates that user-based collaborative filtering struggled with accurate predictions. One of the reasons could be because the method struggles with sparse user-item interaction data (many users have not rated many items), hence fails to identify common grounds between the users. Another reason could be cold start problems (has trouble identifying relevant similarities between users) which affects the performance. There is an increased risk of this model overfitting to specific user profiles, which can lead to significant errors when the model fails to generalize to new users or unseen data.
2. **Item-based Collaborative filtering (RMSE = 0.7710, MAE = 0.7709)**
   This model shows comparatively better results performance and a substantial improvement over UBCF with much lower error rates. This could be attributed to the consistent relationships between items which allows the method to yield better ratings.
3. **Singular Value Decomposition (RMSE = 0.7506, MAE = 0.5932)**
   As compared to the other models, SVD shows the lowest error rates. This is because SVD performs dimensionality reduction and points out the hidden patterns in the data which is effective in improving the accuracy. SVD doesn't explicitly rely much on the similarities between the users or items but rather focuses on the latent factors that can explain the complex patterns in the user behavior and item preference better.

4.1.2 Performance Analysis

We evaluated the performance of the recommendation model based on the 2 key evaluation metrics - Precision and Recall which help in providing insights into the model's abilities to make relevant and complete recommendations. With high precision and recall values, it proves its efficiency in curating the actual recommendations. This hybrid model delivers the best recommendations which demonstrates its ability to discover not only the relevant items but also filter out the irrelevant ones. Collaborative filtering contributes contextual information based on the user-item similarities whereas matrix factorization focuses on pointing out the underlying meaning of the user-item pattern.

4.2 Critical Evaluation

The results educated us with answers to our experimental questions and demonstrated the limitations and opportunities of the recommendation model.

1. **Limitation of Collaborative filtering model**
   We can see that data sparsity impacted the models negatively by reducing the meaningful overlap between users. Cold start problem was a major issue faced by collaborative filtering due to the overdependency on interaction data. This acted as a limitation of the collaborative filtering method to achieve higher accuracy in recommendations.
2. **SVD's Strength**
   The SVD model, however, surpassed the previous model metrics by extracting the underlying factors and providing reliable recommendations thus improving the accuracy. This explains the model's efficiency on the sparse dataset as it generalizes across the dataset. SVD's strength is achieving balanced novelty and relevance using 200 latent factors and addressing the cold start problem faced in the collaborating filtering method. The dimension reduction technique is computationally complex, but it outperformed the other models.
3. **Hyrbid Model Performance**
   This implementation of this hybrid model offered a deeper analysis of the limitations of the traditional methods. We were able to achieve a better recommendation model which is scalable with larger datasets and prove the hybrid model's strength to cater to recommend items that are both relevant and diverse. This approach boosts personalization as well as generalization.
4. **Future Scope**
   While this model provided better results, the model's efficiency can be boosted with addi-

tional contextual data (eg movie metadata) to improve the user satisfaction and accuracy of recommendations. The datasets have fewer attributes which lacked the actual factors that might be the reason for the user to choose a particular movie.

5 Conclusion

This analysis illustrates the efficiency of leveraging a hybrid model with Collborating filtering and Matrix factorization techniques for making relevant recommendations on a sparse dataset. This evaluation highlighted critical trade-offs in the scalability and accuracy of the recommendation using this hybrid model. The results of the study reinforced the superior performance of the matrix factorization model over traditional CF models. This hybrid approach can balance the strengths and weaknesses of both techniques which is crucial while dealing with massive datasets.

While SVD or IBCF performed better, these results come at high computational costs. Currently, the model does not allow real-time recommendation. To overcome these limitations below future directions can be adopted:

- Exploring advanced techniques like Reinforcement learning or online learning models for real-time recommendation study could be worthwhile as a part of future investigation.
- Due to the sparsity in data and reliance on historical data, there is a higher possibility of recommendations reinforcing popular trends. Adding a mechanism that can efficiently handle biases and provide a balanced recommendation is therefore one of the essential future scopes of this model.
- Autoencoders can be explored for better results in latent feature extraction.
- Mechanisms can be built that can interpret the recommendation made by the model could benefit in enhancing user trust and attract users to use the system.

This study analyses the roles of diversity in recommendations. Overemphasis on highly similar item while ignoring the data diversity might result in missing unique factors. In summary, this analysis underscores the benefits of a hybrid model for an efficient recommendation system and its potential to tackle the data sparsity and prediction accuracy issues. Along with the enhancement, it also highlighted the future scope of this model in real-time adaptability, handling bias, etc which boosts the robustness of the model

References

[1] 1. Bennett, James, and Stan Lanning. "The netflix prize." In Proceedings of KDD cup and workshop, vol. 2007, p. 35. 2007.

[2] Tak´acs, G´abor, Istv´an Pil´aszy, Botty´an N´emeth, and Domonkos Tikk. "Matrix factorization and neighbor based algorithms for the netflix prize problem." In Proceedings of the 2008 ACM conference on Recommender systems, pp. 267-274. 2008.

[3]A. Toscher, M. Jahrer, The BigChaos Solution to the Netflix Grand Prize. AT&T Labs, New Jersey, 2009

[4]M. Ali Ghanzanfar, A. Prugel-Bennett, The Advantage of Careful Imputation Sources in Sparse Data-Environment of Recommender Systems: Generating Improved SVD-based Recommendations. School of Electronics and Computer Science, UK, 2013.

[5]Gower, Stephen. "Netflix prize and SVD." University of Puget Sound (2014).

GitHub Link

https://github.ncsu.edu/nmaruva/engr-ALDA-Fall2024-P32