# Movie Recommendation Systems

Harika Rallapalli; Vedant Shaileshbhai Thakkar; Kenneth Zhang
CSC 577 – Recommender Systems, 2024-25 Winter Term, DePaul University

## Abstract

Movie recommendation systems are important in order to help in predicting and recommending a movie which the user will most likely will like. In this paper we have tried to build personalized movie recommendation systems for users. This paper emphasizes exploring various methods for building recommendation systems. The Logistic Regression model, while effective in ranking movies, struggles to distinguish between relevant and irrelevant films, as indicated by its moderate ROC-AUC score, suggesting a need for more advanced techniques to improve its discriminative power.The Bayesian personalized ranking for ranking recommendations based on user interaction has shown a NDCG score of 0.9440 and NDCG with the feature item got a score 0.8987. The model without the use of additional features was proven to have a better ranking for recommendations as having features might disrupt the ranking process.The Hybrid Neural Collaborative Filtering (NCF) model for recommendation systems achieved an NDCG score of 0.7597, demonstrating strong performance in ranking recommendations. The model also achieved an RMSE of 0.2024, indicating good prediction accuracy. This hybrid approach, which combines collaborative filtering with metadata features, highlights the potential of integrating additional features to improve recommendation quality.

**Keywords:** Recommendation system, content based recommendation system, Logistic regression, bayesian personalized ranking, BPR, matrix factorization, Neural Collaborative Filtering (NCF), Hybrid Neural Collaborative Filtering (NCF)

## I. Introduction

In today's world, we have a lot of streaming platforms like Amazon, Hulu, Netflix where we already have millions of users using these platforms. The popularity of these streaming platforms is increasing more in recent years especially after covid. Most of the users on these platforms rely on recommended movies to watch. It's important to have personalized recommendations based on a top list of movies recommended for each user using various features like genre, timestamp, and so on. In this project we will be focusing on designing various personalized recommendation systems to users.

## II. Background

In this paper [2] Content-based filtering struggles with insufficient item descriptions and scalability issues were discussed. My logistic regression, which uses techniques like TF-IDF and hyperparameter tuning, can be utilized to overcome some of the issues faced by content-based filtering, particularly related to insufficient descriptions and scalability.

Recommendation systems are very prevalent in our everyday lives and often provide the correct recommendations based on user preference. Bayesian Personalized Ranking was one of the algorithms that stood out. There are many types of BPR algorithm, such as using top-n recommendations or KNN recommendation algorithms as shown in Milogradskii's research [6]. Additionally, Hu worked on Bayesian personalized ranking based on multiple-layer neighborhoods to focus on hidden information with a new BPR method [3]. The Bayesian Personalized Ranking system works very well with implicit data and I wanted to test out the different optimizations to see whether it would benefit the model in ranking predictions. In the data that was provided, there was no direct implicit data found such as clicks, views, or purchases of movies

Neural Collaborative Filtering (NCF) has emerged as a powerful approach to address the limitations of traditional collaborative filtering methods. He et al. [7] introduced the NCF framework, which leverages neural networks to model user-item interactions, capturing non-linear relationships that traditional matrix factorization methods often

miss. This approach has shown significant improvements in recommendation accuracy, particularly in scenarios with sparse data. Ibrahim et al. [8] extended this work by proposing an intelligent hybrid NCF approach, combining collaborative filtering with content-based features like metadata (e.g., genre, budget, and runtime). Their work demonstrated that integrating metadata enhances recommendation quality, addressing challenges such as the cold-start problem and improving personalization. These studies highlight the potential of hybrid NCF models in delivering accurate and scalable recommendations, making them a promising direction for modern recommendation systems.

## III. Methodology

### A. Content based recommendation using logistic regression

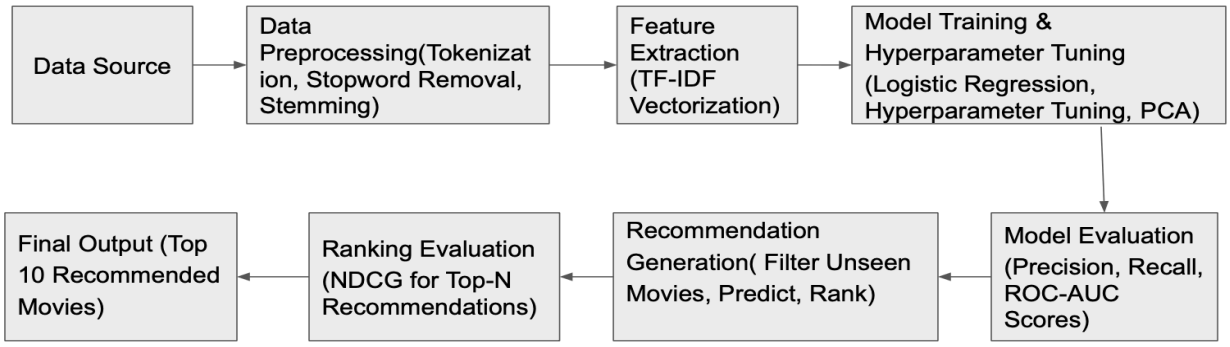


Figure 1 : Basic logistic regression recommendation model overview

Libraries such as sklearn (for model training, evaluation, and hyperparameter tuning), nltk (for text preprocessing), and numpy (for numerical operations) were used in building and optimizing the recommendation system. The recommendation system is built by first cleaning and preprocessing the movie content, using techniques like text tokenization, stop word removal, and stemming. The movie descriptions and genres are combined into a single content column.These combined features are then transformed into numerical features using TF-IDF vectorization, which is limited to the top 5000 features for efficiency. A binary classification model (Logistic Regression) is trained to predict whether a user will like a movie based on its content, with hyperparameter tuning performed via GridSearchCV to optimize the model's performance. GridSearchCV was used to find the best values for two key hyperparameters: C (regularization strength) and solver (optimization algorithm). The optimal parameters for Logistic Regression without PCA were C = 10 and solver = 'newton-cg', while with PCA, the best parameters were C = 1 and solver = 'newton-cg'. The ratings are classified as "liked" ($\geq$3.5) or "not liked" (<3.5). Training and testing are done by splitting the data into an 80% training set and 20% test set using train_test_split. Principal Component Analysis (PCA) is applied to reduce dimensionality for faster and more efficient model training. However, the final recommendation model I have chosen Logistic Regression without PCA because it outperformed the PCA-based model, showing higher Precision (0.7017 vs.0.6941) and ROC-AUC (0.6192 vs. 0.6096), despite almost similar Recall scores (0.7858 vs.0.7890).In my recommendation model I have used predicted probabilities to suggest movies to users, and the ranking quality is evaluated using Normalized Discounted Cumulative Gain (NDCG). Hence, I have done the overall process in a way that combines content-based features, model optimization, and personalized recommendations to effectively predict user preferences.

### B. BPR

The overall objective of the Bayesian Personalized Ranking is to find the maximum probability of a user choosing an item that they have previously interacted with. In the preprocessing for the given movie data, I worked directly with the columns of ‘userId’, ‘movieId’, ‘rating’, and ‘popularity’. Bayesian Personalized ranking takes in a matrix based on user-item interaction which is labeled by R: U x I where R is the rating from the user, U is the list of all the

users, and I for the interactions with the items (Geeks for Geeks). For example if there were movies on a site and giving recommendations, generally a 1 would represent if a user interacted with the movie such as clicking on it or viewing it and 0 would be no interaction at all. Specifically in my data, I focused on the ratings information because there was no implicit data recorded in the dataset. I chose to create implicit data based on the ratings information.In this case, values that are: Greater than or equal 3.5 rating = 1 and less than 3.5 rating = 0

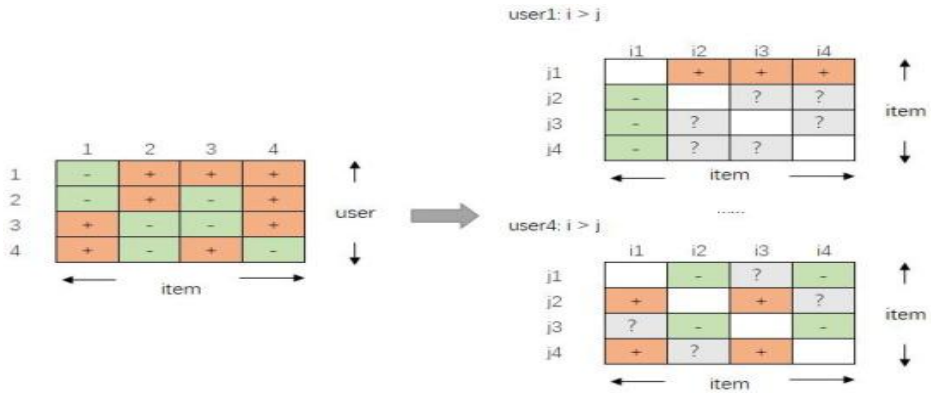


Figure 2 : Matrix Factorization [3]

I utilized the LightFM model [4] with the hyperparameter tuning to find the best AUC score. Given a split data of train and test (80-20), I trained the model on the training dataset using the user-item matrix and the featured items, then tested for the AUC score after on the test set. I performed multiple iterations for the training loop and was able to come out with the best possible tuning for the dataset. I also wanted to test out the difference with or without featured items to see if it would help with the dataset. I also utilized the reference code that was provided by loomlike [5], the code gave me a general basis on how to tune my LightFM BPR model and a general idea on how to train the model along with different parameters.

**C. Hybrid NFC**

The Neural Collaborative Filtering (NCF) and Hybrid NCF models were implemented to enhance traditional collaborative filtering using deep learning. The NCF model combines user and item embeddings through a neural network to predict ratings, while the Hybrid NCF model extends this by incorporating metadata features like genre, budget, and runtime for better recommendations. During preprocessing, user-item interactions and metadata were extracted, with user and item IDs encoded for embedding layers. Metadata features were normalized and combined with interaction data to create a comprehensive input for the hybrid model.

The base NCF model uses embedding layers and a neural network, trained with binary cross-entropy loss and the Adam optimizer. The Hybrid NCF model concatenates metadata features with user and item embeddings, passing them through dense layers with ReLU activation and dropout for regularization. The dataset was split into training (80%) and testing (20%) sets, and hyperparameter tuning was performed using Keras Tuner to optimize embedding size, learning rate, and dropout rate. Regularization techniques like L2 regularization and dropout were applied, and early stopping was implemented to prevent overfitting.

The models were evaluated using RMSE, MAE, Precision, Recall, ROC-AUC, and NDCG. The Hybrid NCF model achieved an NDCG score of 0.7597 and an RMSE of 0.2024, demonstrating strong ranking performance and decent prediction accuracy. By combining collaborative filtering with metadata features and deep learning, the Hybrid NCF model provides a scalable and robust solution for personalized recommendations, showcasing the potential of advanced techniques in modern recommendation systems.

## IV. Results

### A. Data collection

The data is collected from Kaggle website [1] which consists of movielens dataset that has been collected by the Group Lens Research project. The dataset consists of a total of 45,000 movies and includes various features such as genre, movie overview and so on. This dataset consists of 26 million ratings obtained from 270,000 users. Ratings are given on a scale of 1 to 5 where 1 is the lowest rating and 5 is the highest rating. Also it can be noted that there are no missing ratings in our data. Before we performed our analysis we made sure the data is cleaned by performing various techniques like converting column data types when needed especially for id columns, filling missing values for example in case of missing string value we have filled with empty string value and we also removed any outliers in the dataset. The dataset was limited to 100k in order to avoid system crashing as we have 26 million ratings. For all the models, the dataset is split as 80 training sets and 20 testing sets. In case of classification we have used greater than equal to 3.5 rating as liked (1) or not liked (0). All the models perform recommendations of top 10 movies based on the models output for a user.

### B. Logistic regression

The logistic regression model predicted whether a drama was liked based on its content and rating. After preprocessing (cleaning the text, using TF-IDF, and applying PCA), hyperparameter tuning was done via GridSearchCV. The results showed that the best parameters for logistic regression without PCA were: C = 10, solver = 'newton-cg'. The performance metrics for logistic regression without PCA were:

- Precision: 0.7017
- Recall: 0.7858
- ROC-AUC: 0.6192
- RMSE: 0.4727
- MAE: 0.4097
- NDCG: 0.7281

For logistic regression with PCA, the best parameters were C = 100, solver = 'liblinear'. The performance metrics with PCA were:

- Precision: 0.6941
- Recall: 0.7890
- ROC-AUC: 0.6096
- RMSE: 0.4758
- MAE: 0.4119
- NDCG: 0.6020

PCA slightly reduced the model's performance (lower ROC-AUC, precision, and NDCG). Thus, it was not included in the final recommendation function to predict top movies for users. The model using Logistic Regression (without PCA) was used where TF-IDF was used to represent movie content, and predictions were made for unseen movies based on user preferences. The model successfully ranked movies by predicted relevance for the user id 41507, yielding strong recommendations (NDCG@10 = 1.0000), while bad recommendation for failure case shows in ascending order of predicted probability generated intentionally poor recommendations.

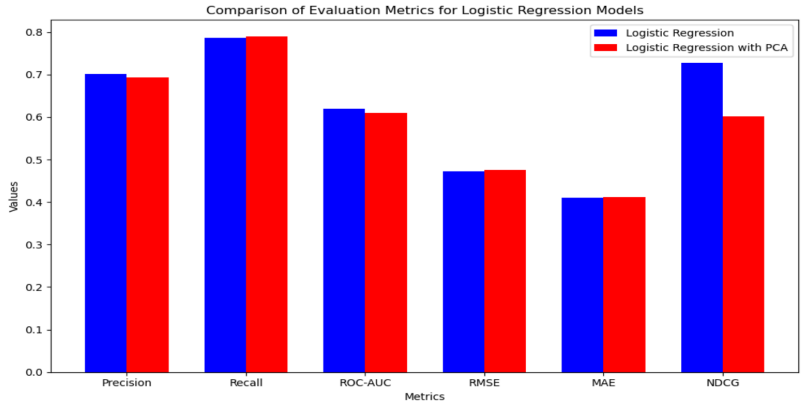


Figure 3: Precision, recall, ROC-AUC, RMSE, MAE, NDCG for logistic regression model and logistic regression model with PCA

```
Top Movie Recommendations for User 41507:
Legionnaire | Predicted Rating: 0.9814
Charlie's Angels | Predicted Rating: 0.9805
Titan A.E. | Predicted Rating: 0.9800
The Tailor of Panama | Predicted Rating: 0.9795
The Ghost of Frankenstein | Predicted Rating: 0.9791
Goodbye Bafana | Predicted Rating: 0.9762
Weekend at Bernie's II | Predicted Rating: 0.9757
Lassie Come Home | Predicted Rating: 0.9755
The Final Countdown | Predicted Rating: 0.9748
Exodus | Predicted Rating: 0.9747

NDCG@10 for User 41507: 1.0000
```

```
Bad Movie Recommendations for User 37422:
 My Super Ex-Girlfriend | Predicted Rating: 0.1986
 The Lady from Shanghai | Predicted Rating: 0.2003
 Ladybug Ladybug | Predicted Rating: 0.2474
 A Date with Judy | Predicted Rating: 0.2568
 The Weather Man | Predicted Rating: 0.2691
 A Streetcar Named Desire | Predicted Rating: 0.2760
 Muriel's Wedding | Predicted Rating: 0.2782
 The Key to Reserva (La clave Reserva) | Predicted Rating: 0.2830
 House of 1000 Corpses | Predicted Rating: 0.3009
 Licence to Kill | Predicted Rating: 0.3078
```

Figure 4: Good example for our logistic regression recommendation along with NDCG score and Bad example for our logistic regression recommendation system

**C. BPR**

After tuning the parameters for the LightFM model on the test set, I was able to obtain the parameters of Number of Components: 96, Learning Rates: 0.001, Epochs: 90, Item Alpha: 0.005, and User Alpha: 0.001. The results for precision came out to 0.1837 and recall came out to 0.0726 for the test set.

Finally, the overall evaluation after training and testing the model came out to: Precision: 0.9124, Precision with Feature: 0.8981, Recall: 0.0724, Recall with Feature: 0.0712, NDCG: 0.9440, NDCG with Feature: 0.8987.

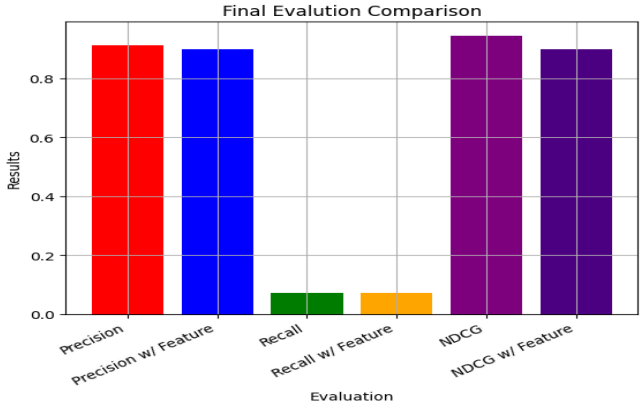


Figure 5: BPR Evaluation Results

In the final results, there was a marginal difference between the precision and recall with the addition of the popularity feature item. The more noticeable distinction in the results comes from the NDCG. Although it was not a huge difference, the scores are more observable from the graph. The scores for precision came with a difference of around 0.02, recall had a difference of about 0.001, and NDCG had a larger difference of around 0.05.

```
Recommended movies without Feature Item for user 150:
1. "The 39 Steps" with Score: 0.0239
2. "Rope" with Score: 0.0230
3. "Bridge to Terabithia" with Score: 0.0230
4. "Terminator 3: Rise of the Machines" with Score: 0.0229
5. "A Nightmare on Elm Street" with Score: 0.0211
6. "Young and Innocent" with Score: 0.0208
7. "The Million Dollar Hotel" with Score: 0.0206
8. "The Conversation" with Score: 0.0201
9. "Men in Black II" with Score: 0.0178
10. "Light of Day" with Score: 0.0145
```

```
Bad recommended movies without Feature Item for user 150:
1. "Monty Python and the Holy Grail" with Score: -0.0176
2. "Cat on a Hot Tin Roof" with Score: -0.0150
3. "In Time" with Score: -0.0148
4. "Longitude" with Score: -0.0146
5. "Eyes Wide Shut" with Score: -0.0145
6. "Shuang ma lian huan" with Score: -0.0144
7. "The Good Thief" with Score: -0.0137
8. "The Dark" with Score: -0.0135
9. "The Sixth Sense" with Score: -0.0133
10. "Indestructible Man" with Score: -0.0133
```

Figure 6 : Good BPR outputs (Left) and Bad BPR outputs (Right)

**D. Hybrid NFC**

The journey of implementing and refining the Neural Collaborative Filtering (NCF) and Hybrid NCF models was both challenging and rewarding. Starting with the base NCF model, a Mean Absolute Error (MAE) of 0.1244 was achieved, which provided a solid foundation for further improvements. The base NCF model leveraged user-item interaction data to predict user preferences, demonstrating the potential of neural networks in collaborative filtering tasks. This initial result set the stage for exploring more advanced techniques to enhance the model's performance.

To improve the base NCF model, hyperparameter tuning was performed, focusing on optimizing parameters such as embedding size, learning rate, and dropout rate. This process resulted in a validation MAE of 0.1214 for the Hyperparameter-Tuned NCF model, showcasing the importance of fine-tuning model parameters to achieve better accuracy. The reduction in MAE highlighted how small adjustments to the model architecture and training process could lead to measurable improvements in prediction quality.

Building on the success of the hyperparameter-tuned NCF model, the Regularized Hybrid NCF model was developed. This model incorporated metadata features such as genre, budget, and runtime, combining collaborative filtering with content-based features. The inclusion of metadata allowed the model to capture additional contextual information about movies, leading to an RMSE of 0.2015. This improvement demonstrated the benefits of integrating diverse data sources to enhance recommendation systems. However, to address overfitting, the Early Stopping Model was implemented, which achieved a slightly higher RMSE of 0.2042 but maintained competitive performance.

The final Hybrid NCF model combined the strengths of collaborative filtering and metadata features, achieving an RMSE of 0.2024 and an MAE of 0.1623. These metrics indicated strong prediction accuracy, while the model's Precision of 0.5061, Recall of 0.6101, and ROC-AUC of 0.557

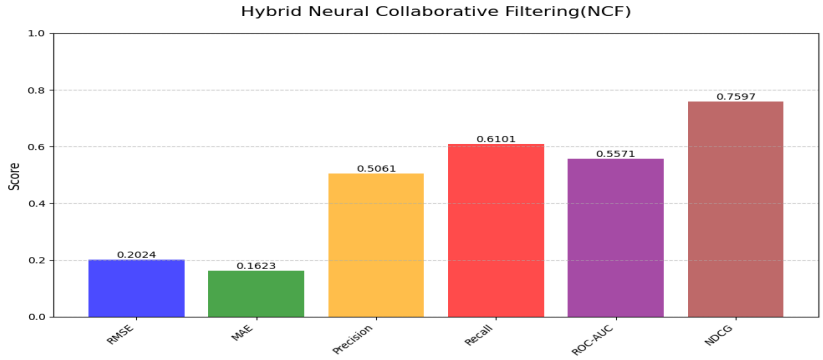


Figure 7: RMSE, MAE, Precision, recall, ROC-AUC,NDCG of Hybrid NCF

Recommendations generated by the final Hybrid NCF model**:**

```
Top-10 Recommendations for User 0:
Movie: Boogie Nights, Predicted Rating: 0.7310
Movie: Sweet Sixteen, Predicted Rating: 0.7235
Movie: Wag the Dog, Predicted Rating: 0.7199
Movie: And Then There Were None, Predicted Rating: 0.7190
Movie: A Bridge Too Far, Predicted Rating: 0.7145
Movie: Sister Act, Predicted Rating: 0.7142
Movie: Jack & Sarah, Predicted Rating: 0.7130
Movie: Ghost Dog: The Way of the Samurai, Predicted Rating: 0.7124
Movie: Aliens vs Predator: Requiem, Predicted Rating: 0.7123
Movie: Cat on a Hot Tin Roof, Predicted Rating: 0.7119
```

Figure 8: Top 10 recommendations

**E. Models Result Comparison (BPR, BPR with feature, Logistic Regression, Logistic Regression with PCA, Hybrid NCF) :**

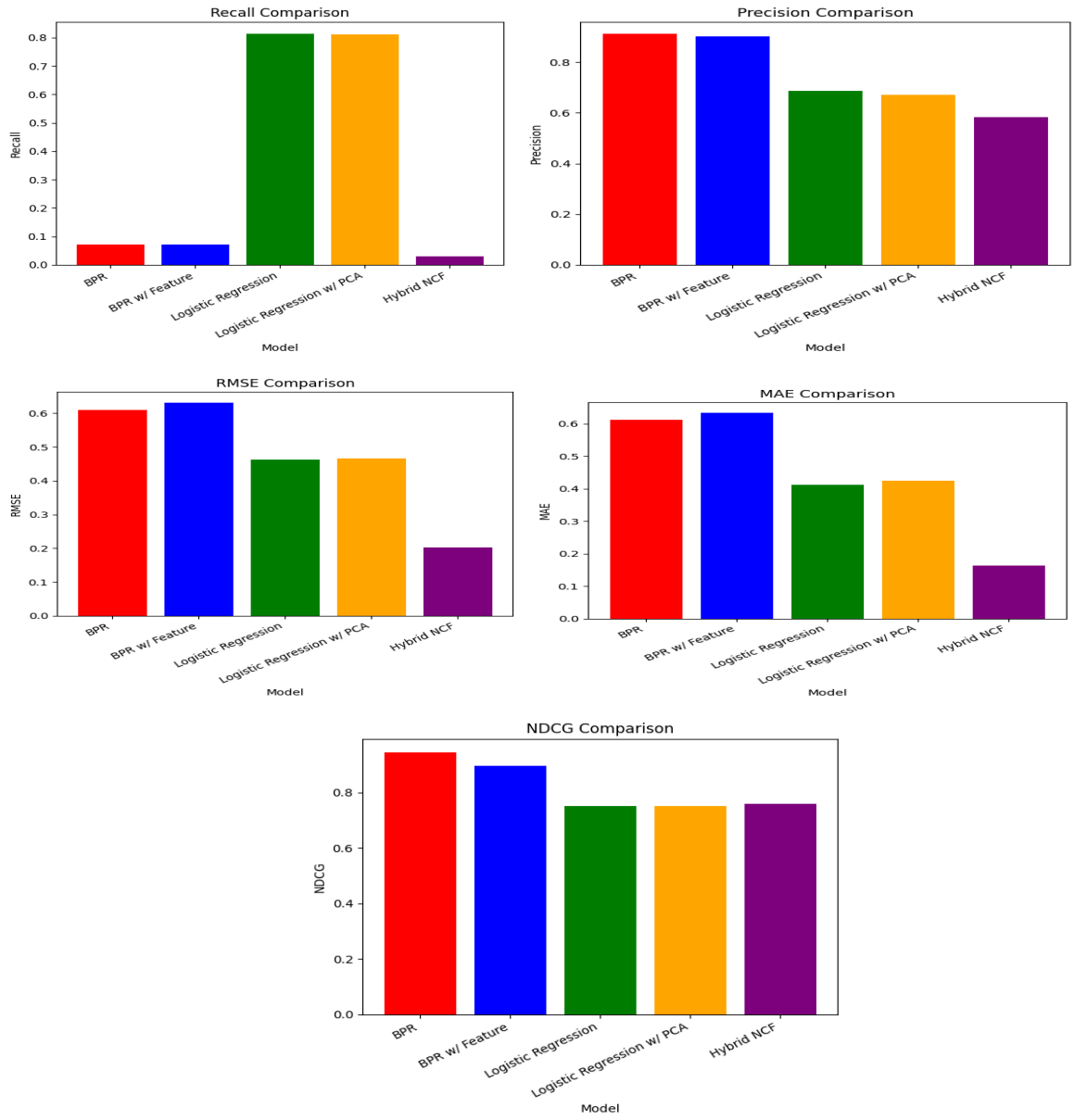


Figure 9 : Recommendation system models comparison using  precision, recall, RMSE, MAE and NDCG

Here in the final results, logistic regression had the best overall recall, BPR had the best precision, NCF had the best RMSE and MAE, and logistic regression had similar scores in NDCG with the NCF model. Some of the models had similar scores, but the overall best model was the logistic regression with moderate scores throughout all of the evaluations.

## V. Conclusion

In this project, we successfully explored multiple recommendation models, including Logistic Regression, Bayesian Personalized Ranking (BPR), and Hybrid Neural Collaborative Filtering (NCF), each showing varying levels of performance. Hyperparameter tuning with GridSearchCV improved Logistic Regression's models performance, while dimensionality reduction using PCA caused a slight drop in model's performance. Despite its effectiveness in ranking movies with NDCG score of 0.7281, the model had limitations in distinguishing between relevant and irrelevant movies, suggesting that future improvements could involve non-linear models, hybrid approaches, and enhanced feature engineering. Exploring dimensionality reduction techniques like t-SNE could potentially improve this model's results.The BPR models achieved excellent NDCG scores, with the model without additional feature items performing slightly better, though future work could involve tackling the cold start problem and leveraging implicit data, such as clicks and views. The hybrid NCF model combined user-item interactions with movie metadata, achieving a strong NDCG score and demonstrating effectiveness in personalized recommendations, with future work focusing on advanced architectures and addressing cold start issues. Overall, the hybrid NCF model proved to be a powerful and scalable solution for personalized movie recommendations.To further enhance recommendation quality for all our models, the integration of multimodal data, such as reviews, audio, or video content, could be explored.

## References


1. Kaggle movies dataset, https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset.

2. Lops, Pasquale & de Gemmis, Marco & Semeraro, Giovanni. (2011). Content-based Recommender Systems: State of the Art and Trends. 10.1007/978-0-387-85820-3_3. https://www.researchgate.net/publication/226098747_Content-based_Recommender_Systems_State_of_the_Art_and_Trends

3. Hu, Yutian, et al. "Bayesian Personalized Ranking Based on Multiple-Layer Neighborhoods." Information Sciences, Elsevier, 4 July 2020, www.sciencedirect.com/science/article/pii/S0020025520306563.

4. Kula, Maciej. "LIGHTFM." LightFM - LightFM 1.16 Documentation, 2016, making.lyst.com/lightfm/docs/lightfm.html.

5. loomlike. "Cornac_bpr_deep_dive.ipynb." GitHub, 2023, https://github.com/recommenders-team/recommenders/blob/main/examples/02_model_collaborative_filtering/cornac_bpr_deep_dive.ipynb.

6. Milogradskii, Aleksandr, et al. "Revisiting BPR: A Replicability Study of a Common Recommender System Baseline." Revisiting BPR: A Replicability Study of a Common Recommender System Baseline, 21 Sept. 2024, arxiv.org/html/2409.14217v1.

7. He, Xiangnan, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. "Neural collaborative filtering." In *Proceedings of the 26th international conference on world wide web*, pp. 173-182. 2017.

8. Ibrahim, Muhammad, Imran Sarwar Bajwa, Nadeem Sarwar, Fahima Hajjej, and Hesham A. Sakr. "An intelligent hybrid neural collaborative filtering approach for true recommendations." *IEEE Access* 11 (2023): 64831-64849.