CS657 December 3, 2024

HW4:Recommender System

Antara Tewary (G01413546),Ankit Kumar (G01436204)

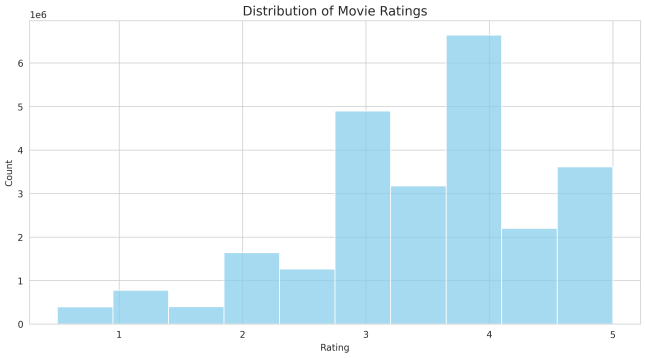
1. **Introduction**

   This assignment explores three recommendation algorithms using MovieLens 25M dataset which has 25 million ratings for 62,000 movies from 162,000 users. The dataset's rich information about user's preferences and movie attibutes gives us an opportunity to look into the trends and patterns in data. We go through three stages: starting with Alternating Least Squares (ALS) matrix factorization, optimized by cross validation, and enhancing it with item-item collaborative filtering for a hybrid approach. At the end we add a supervised learning componend that uses movie features, which creates a robust three component hybrid system.

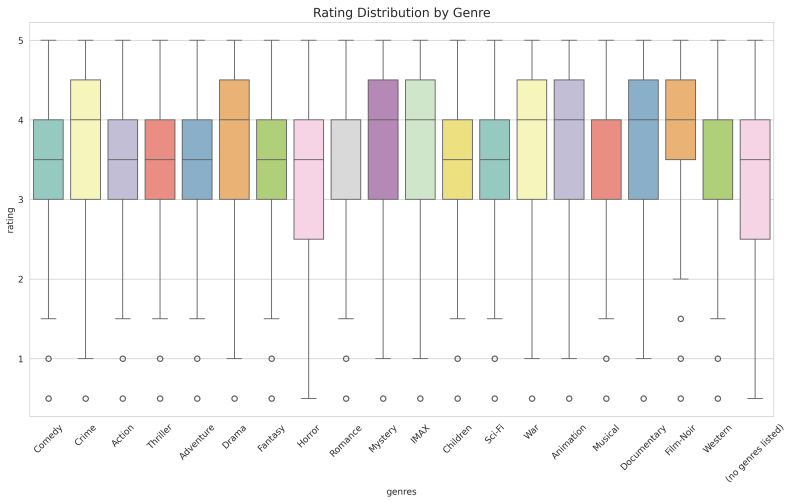
   We implement this in Spark and focus on optimizing performance within cluster's constraints. Effectiveness of our approach is evaluated using RMSE, MSE, and MAP metrics which ensures a thorough comparison of each approach.

2. **Data Analysis**

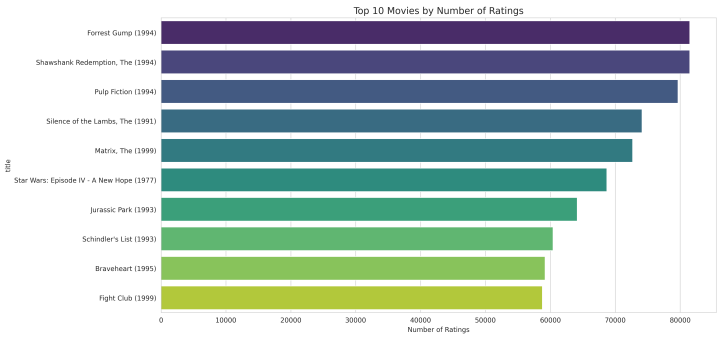
   - *Dataset Overview:* Ratings data contains userId, movieId, and rating fields, while movies data includes movieId, title, and genres. A positivity bias in ratings is evident in Figure 1(a), with a median of 4.0 stars and most ratings clustered in the 3-4 star range.
   - *Data Cleaning:* We converted column types (IDs as integers, ratings as floats) and repartitioned into 12 partitions for efficiency. We cached frequently accessed dataframes, and implemented a drop strategy for cold start scenarios in ALS model.
   - *Exploratory Data Analysis:* As shown in Figure 1(e), classics like `Forest Gump` and `The Shawshank Redemption` dominate the ratings, each exceeding 70,000. For genre, `Drama` is the most popular, followed by `Comedy` (Figure 1(d)). Specialized genres like `Film-Noir` and `IMAX` have fewer ratings. As seen in Figure 1(b), most genres' median ratings are between 3-4 stars. `Documentaries` and `Dramas` have slightly higher median ratings, while `Horror` shows the widest variance.
   - *Data Partitioning:* The dataset was split 80-20 for training (19,999,186 ratings) and validation (5,000,909 ratings) using Spark's randomSplit with seed 42. Larger validation sets ensure robust evaluation. Figure 1(e) shows the scatter plot of average ratings versus number of ratings per movie, revealing a characteristic pattern where rating variance decreases as the number of ratings increases, suggesting more reliable average ratings for frequently-rated movies. This motivated us to use "drop" strategy for cold-start scenarios in the collaborative filtering approaches.



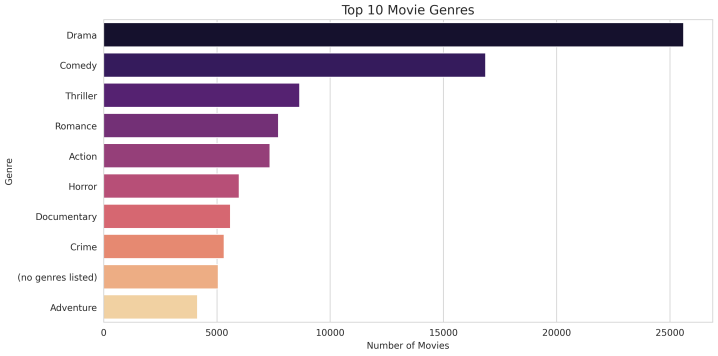
(a) Rating Distribution



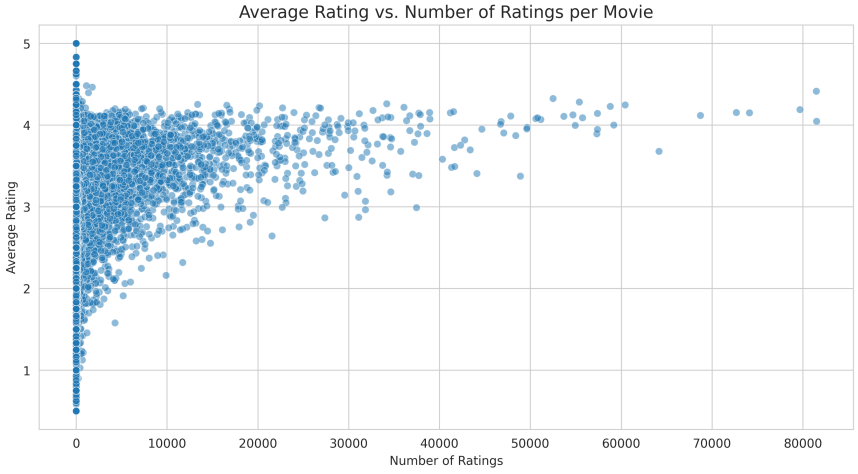
(b) Histogram of Ratings



(c) Movie popularity



(d) Genre Distribution



(e) Rating vs count

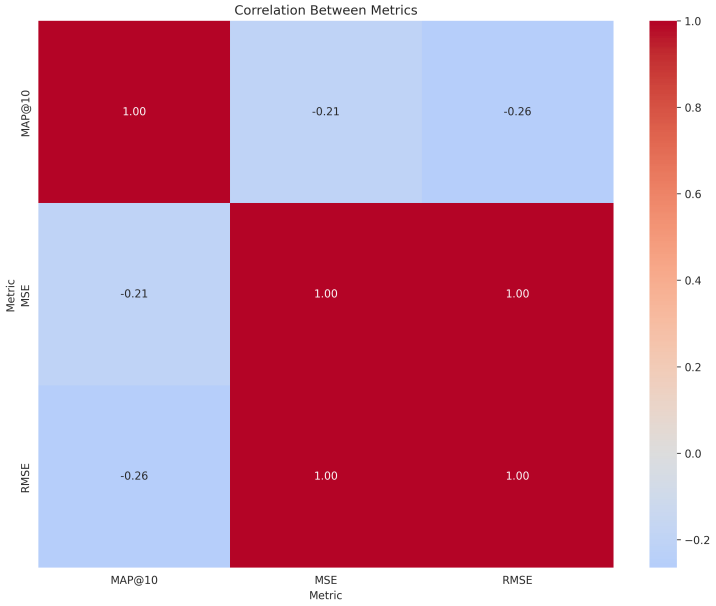
3. Implementation

- *Base ALS Recommender System:*We first implemented Spark's ALS algorithm as our base. This model is configured with nonnegativity constraints and a drop strategy for cold-start scenarios, ensuring robustness. We implemented Cross-Validation using a Parameter Grid that explores rank values of 10 and 20, iteration counts of 5 and 15, and regularizatin parameters of 0.01 and 0.1.We distributed data in 100 partitions to balance processing load.
- *Hybrid System with ALS and ITem-Item Collaborative Filtering:*The hybrid system enhances the basic ALS model by introducing item-item collaborative filtering. We calculated movie similarities using cosine similarity and integrate it with ALS predictions through a weighted approach. We explored different weight combinations(0.3,0.5,0.7) with complementary weights assigned to the CF component. This method leverages the strength of global patterns captured by ALS and local similarities identified by item-item CF.
- *Enhanced Hybrid System with Supervised Learning:*The final stage introduces a supervised learning component through Random Forest regressor, which creates the three-way hybrid architecture. This component introduces movie features through feature engineering, focusing on genre and metadata. Our `EnhancedHybridRecommender` class explores various weight combinations including balanced (0.4, 0.3, 0.3), ALS-dominant (0.6, 0.2, 0.2), and supervised-learning-dominant (0.2, 0.2, 0.6).

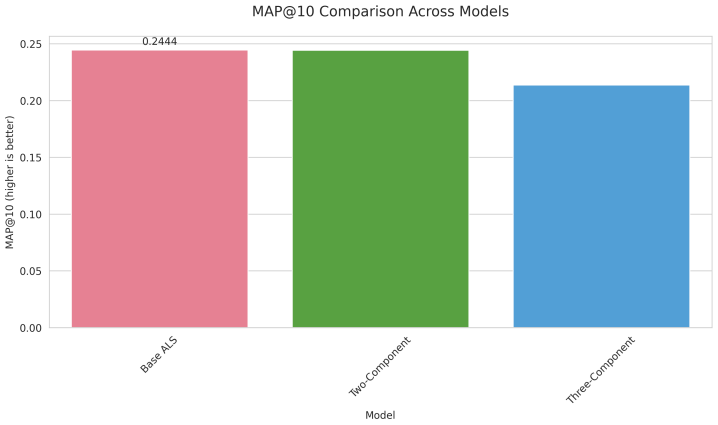
4. Performance Analysis

- ***Evaluation Metrics***
  Model performance is assessed using three metrics. RMSE(Root Mean Square Error) and MSE(Mean Square Error) measure prediction accuracy where lower values indicate better performance. Map10(Mean Avg Precision at 10) evaluates the recommendation quality by assessing how well the system ranks the top 10 recommendations for each user, with higher values indicating better ranking performance.
- ***Visualization Analysis***
  - The correlation analysis between these metrics (Figure 1(f)) reveals a strong positive correlation (1.0) between RMSE and MSE, while MAP@10 shows a weak negative correlation with both error metrics (-0.21 with MSE and -0.26 with RMSE), suggesting that optimizing for prediction accuracy doesn't necessarily improve recommendation ranking quality.

– Figure 1(g) shows Map@10 performance across all three models, showing consistent performance around 0.24. The radar plot(Figure 1(h))visualization demonstrates how each model variant performs across all three metrics, providing a clear view of the trade-offs between prediction accuracy and ranking quality. The distribution of metrics(Figure 1(i)) highlights the relative performance of each approach.

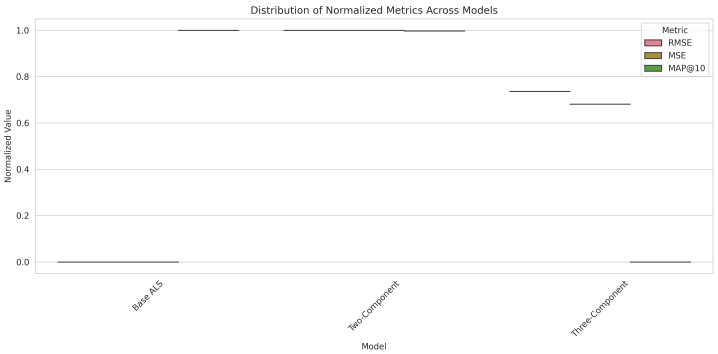


(f) Correlation Between Metrics



(g) MAP@10 Comparison Across Models

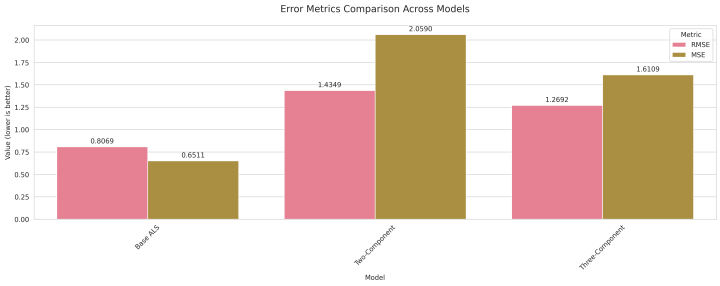
(h) Relative Performance Across Metrics (Radar Plot)



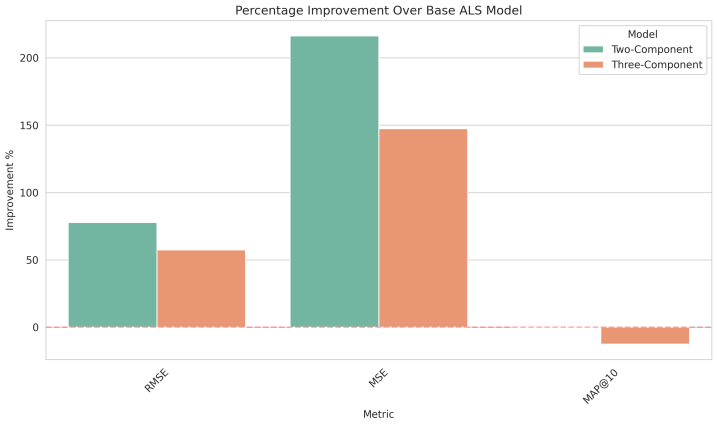
(i) Distribution of Normalized Metrics Across Models

- ***Results and Comparison***
  The performance comparison across our three implemented models reveals interesting trade-offs between prediction accuracy and ranking quality. The baseline ALS model achieved the best error metrics with an RMSE of 0.8069 and MSE of (Figure 1(j)). While the two-component hybrid system showed higher error rates (RMSE: 1.4349, MSE: 2.0590), it maintained comparable MAP@10 performance at 0.2443, suggesting that the addition of item-based collaborative filtering preserved recommendation ranking quality despite increased prediction error. The three-component system demonstrated a balanced performance with improved error metrics (RMSE: 1.2692, MSE: 1.6109) compared to the two-component system, though still not matching the baseline ALS accuracy. Notably, as shown in Figure 1(k), while both hybrid approaches showed increased error metrics compared to the baseline, they maintained stable MAP@10 scores, indicating robust ranking performance. The heatmap visualization (Figure 1(l)) particularly highlights how the three-component system achieves a better balance between prediction accuracy and

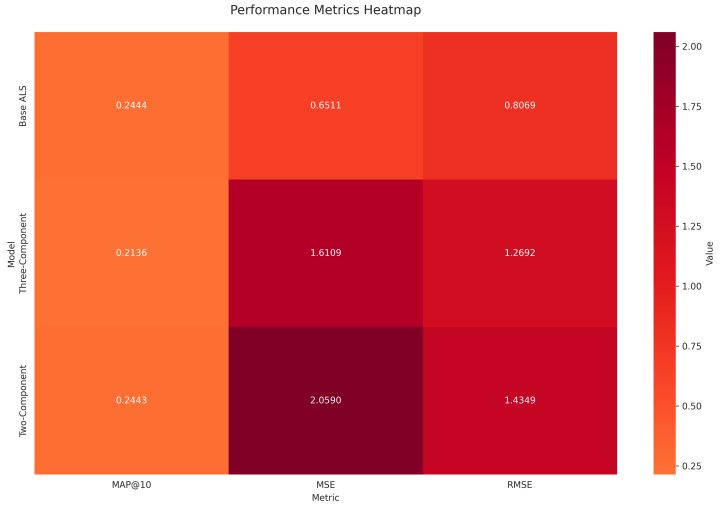
ranking quality, suggesting it might be the most practical choice for real-world applications despite not being the top performer in any single metric.



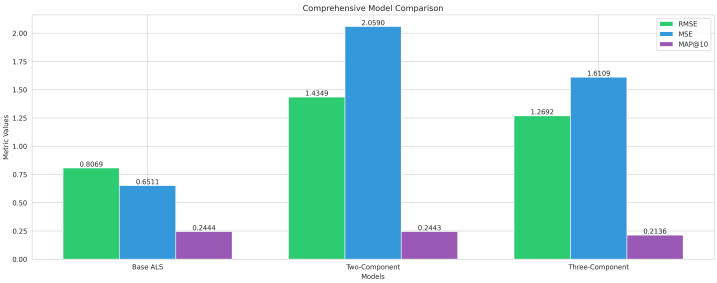
(j) Error Metrics Comparison



(k) Improvement Analysis Across Models



(l) Metrics Heatmap



(m) Model Comparison

5. Challenges And Solutions

The implementation of our recommender system faced three primary challenges:

- memory management with the large-scale MovieLens dataset
- computational efficiency in hybrid model integration
- optimization of evaluation metrics across different model components

Some solutions that we tried:-

- For memory constraints, we partitioned the data into 100 partitions, used caching, and Spark's broadcast varuables for smaller datasets to reduce shuffle operations.
- To integrate multiple recommendation approaches, we used an efficient weighted combination strategy with optimized join operations.
- Performance optimization was achieved through parallel processing of model components and careful configuration of storage levels for intermediate results.

6. Conclusion

As shown in Figure 1(m), our experimentation with three recommender system approaches revealed clear performance patterns. The baseline ALS model achieved the best accuracy metrics (RMSE: 0.8069, MSE: 0.6511), while the hybrid approaches demonstrated higher error rates but consistent MAP@10 scores. The three-component system showed improved error metrics over the two-component hybrid, suggesting better balance between accuracy and ranking.