MovieLens Recommendation System

Your Name

2025-03-07

Contents

1 Movie Recommendation System Report Outline

1.1 1. Introduction/Overview/Executive Summary

1.1.1 Dataset Description

- The MovieLens 10M dataset contains approximately 10 million ratings applied to 10,000 movies by 72,000 users
- The dataset includes user IDs, movie IDs, ratings (0.5-5 stars), timestamps, movie titles, and genres
- Each movie can belong to multiple genres (e.g., "Comedy|Romance|Drama")

1.1.2 Project Goal

- Develop a recommendation system that accurately predicts how users will rate movies they haven't seen yet
- Minimize prediction error using the Root Mean Square Error (RMSE) metric
- Target: Achieve RMSE < 0.86490 on the final holdout test set

1.1.3 Key Steps Performed

- Data preparation and partitioning into training, validation, and test sets
- Progressive model building from simple baseline to complex models
- Feature engineering to incorporate movie, user, and genre effects
- Regularization to prevent overfitting
- Final model evaluation on a separate holdout test set

1.2 2. Methods/Analysis

1.2.1 Data Preparation

- Downloaded and extracted MovieLens 10M dataset
- Joined ratings data with movie metadata
- Created data partitions:
  - 90% for model development (edx)
  - 10% for final evaluation (final_holdout_test)
- Further divided the development data:
  - 80% for training (edx_train)
  - 20% for validation

1.2.2 Data Exploration and Visualization

- Analysis of rating distribution (average ratings, variance)
- Examination of movie popularity and its relationship with ratings
- Analysis of user rating patterns
- Investigation of genre impact on ratings
- Visualization of key relationships and patterns in the data

1.2.3 Insights from Exploration

- Some movies are consistently rated higher/lower than average
- Some users tend to rate more generously/critically than others
- Certain genres receive systematically different ratings
- Movies with few ratings show more extreme average ratings
- Rating patterns differ across genres

1.2.4 Modeling Approach

1. **Baseline Model**: Global mean rating for all predictions
2. **Movie Effects Model**: Adjusted for movie-specific deviations
3. **User Effects Model**: Incorporated user rating tendencies
4. **Combined Model**: Integrated both movie and user effects
5. **Regularized Model**: Added regularization to prevent overfitting
6. **Genre Effects Model**: Included genre-specific biases
7. **Genre-Specific User Effects**: Analyzed user preferences by genre

1.3 3. Results

1.3.1 Model Performance Comparison

- Baseline Model RMSE: [value]
- Movie Effects Model RMSE: [value]
- User Effects Model RMSE: [value]
- Combined Effects Model RMSE: [value]
- Regularized Model RMSE: [value]
- Genre Effects Model RMSE: [value]
- Genre-Specific User Effects RMSE: [value]

1.3.2 Best Model Analysis

- Detailed examination of the best-performing model
- Analysis of regularization parameter (lambda) tuning
- Visualization of prediction accuracy vs. actual ratings
- Discussion of feature importance (movie, user, and genre effects)
- Examples of well-predicted vs. poorly-predicted ratings

1.3.3 Final Model Performance

- RMSE on final holdout test set: [value]
- Comparison to project target (RMSE < 0.86490)
- Analysis of prediction distribution across different rating values

1.4 4. Conclusion

1.4.1 Summary

- The project successfully developed a recommendation system using the MovieLens dataset
- A regularized model incorporating movie, user, and genre effects performed best
- The final model achieved an RMSE of [value] on the holdout test set
- Key factors affecting prediction accuracy were [list factors]

1.4.2 Limitations

- The model doesn't account for temporal effects (e.g., changing user preferences over time)
- New users and movies (cold start problem) would have limited prediction accuracy
- Genre categories are broad and may not capture nuanced content preferences
- The dataset represents user behavior from a specific time period

1.4.3 Future Work

- Incorporate time-based features to capture evolving preferences
- Explore matrix factorization and latent factor models
- Implement content-based features using movie metadata
- Develop hybrid recommendation approaches
- Test the model on more recent datasets
- Incorporate additional features like movie popularity, recency, and user activity