Movie Rating Prediction Analysis

Christoph Hartleb

2024-10-13

Contents

Executive Summary . . . 1
Introduction . . . 1
Data Collection & Preprocessing . . . 1
- Data Sources . . . 1
- Data Preprocessing Steps . . . 1
- Feature Engineering . . . 2

Methodology . . . 2
- Random Forest Model . . . 2
- Model Pipeline . . . 2

Results and Evaluation . . . 3
- Model Performance . . . 3
- Predicted vs Actual Ratings Plot . . . 3
- Insights: . . . 3

Discussion . . . 3
- Insights from Model Predictions . . . 3
- Model Limitations . . . 4

Conclusion . . . 4
Recommendations for Future Work . . . 4
- Model Improvements . . . 4
- Business Application . . . 4

Executive Summary

This report presents the analysis and results of a predictive model developed to forecast movie ratings based on historical user behavior and movie features. Using a Random Forest model, we achieved a low Root Mean Squared Error (RMSE) on a holdout test dataset of approximately:

```
## [1] 0.2041044
```

The results demonstrate the effectiveness of machine learning in building recommendation systems. Key factors contributing to rating predictions include the year of movie release, user average ratings, and years since release.

Introduction

The goal of this analysis is to predict movie ratings using a machine learning model, specifically a Random Forest algorithm, and to assess the accuracy of these predictions. The analysis aims to provide insights into how user preferences and movie attributes affect ratings, which can be used to enhance recommendation

systems. This report documents the end-to-end process, from data collection and preprocessing to model evaluation and performance metrics.

Data Collection & Preprocessing

Data Sources

The dataset used in this analysis is the **MovieLens 10M Dataset**, which contains millions of ratings from users on various movies. The primary datasets include:

- **Movies Data**: Contains movie IDs, titles, and genres.
- **Ratings Data**: Contains user ratings along with timestamps.

Data Preprocessing Steps

Data preprocessing steps were automated through external R scripts sourced directly into this report. Key preprocessing steps include:

- Extracting the release year from movie titles.
- Computing user average ratings from historical data.
- Calculating years since release and ensuring that test data is structured to match the model training data.
- Missing ratings were handled to avoid bias in the predictions.
- The dataset was aligned with the model's training data, ensuring consistency in features.

Feature Engineering

In this stage of the analysis, we enhance the dataset through **feature engineering**, which involves the creation of new variables aimed at improving the prediction accuracy of our model. The following enhancements were made:

1. **Year of Release**:
    - The **Year** of each movie is directly extracted from the movies dataset. This serves as a key feature in understanding trends in movie ratings over time.
2. **Rating Year and Month**:
    - New features, **RatingYear** and **RatingMonth**, were created by converting the `Timestamp` field from Unix time to a human-readable format.
    - This allows for temporal analysis of user ratings, revealing seasonal trends and patterns in user behavior.
3. **Missing Values Check**:
    - A check was implemented to quantify any missing values in the dataset post-feature engineering. This ensures data quality before moving forward in the analysis pipeline.
4. **Data Splitting**:
    - The dataset is divided into training and test sets using an 80/20 split. This is crucial for validating the model's performance on unseen data.
5. **Processed Data Saving**:
    - The processed training and test datasets are saved in RDS format for efficient access in subsequent modeling steps.

Methodology

Random Forest Model

We chose the **Random Forest** algorithm due to its robustness in handling both numeric and categorical data and its ability to minimize overfitting. The model was trained on historical data using features such as:

- **Movie attributes**: Release year, average movie rating.
- **User behavior**: Average rating given by a user, the number of ratings made.

Model Pipeline

The pipeline for our model can be summarized as:

Model Training: Train a Random Forest on a training dataset.

Prediction: Apply the trained model to predict ratings for the holdout test dataset.

Evaluation: Calculate performance metrics such as RMSE and visualize predicted vs actual ratings.

Results and Evaluation

Model Performance

The performance of the Random Forest model was evaluated using **Root Mean Squared Error (RMSE)**, which is a widely used metric in regression problems. The RMSE reflects the standard deviation of the prediction errors, indicating how close the predicted ratings are to the actual ratings.

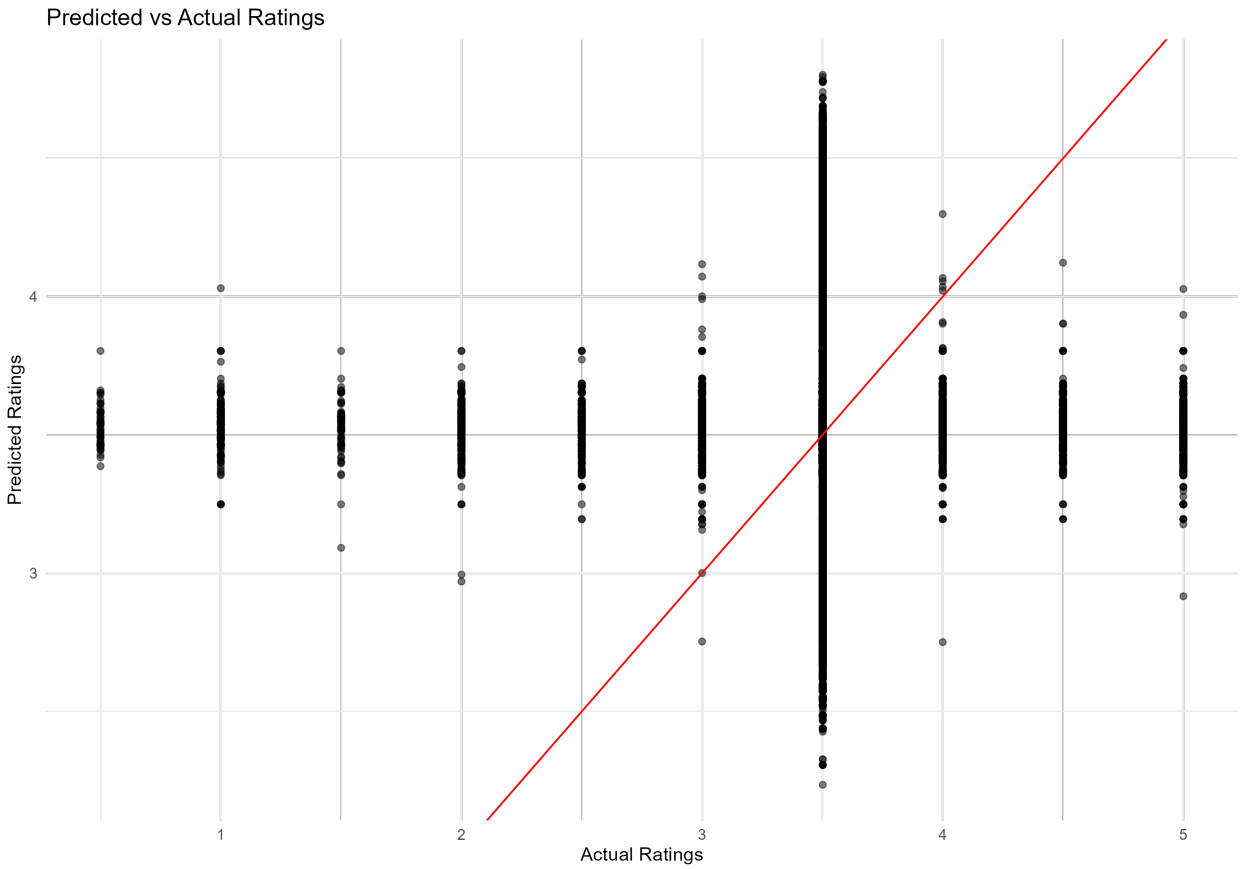
The model achieved an RMSE of

```
## [1] 0.2041044
```

indicating a strong predictive ability on the test data.

Predicted vs Actual Ratings Plot

To further assess the model's performance, we compare the predicted and actual ratings through a scatter plot. Ideally, the predictions should align closely with the actual values, represented by a red diagonal line.



The plot represents the predicted vs. actual ratings from the movie recommendation system:

1. **X-axis (Predicted Ratings)**: These are the ratings generated by the recommendation model based on user and movie data. The predicted ratings range from around 2 to 5, indicating that the model is generating ratings within this range.
2. **Y-axis (Actual Ratings)**: These are the true ratings that users provided for the movies. Like the predicted ratings, these actual ratings also range between 2 and 5.
3. **Red Line**: The diagonal red line represents the ideal scenario where the predicted ratings match the actual ratings perfectly. If a point lies on this line, the model's prediction for that movie is exactly correct.
4. **Data Points**: Each point in the plot represents an individual movie rating. Points that are closer to the red line indicate more accurate predictions, whereas points farther from the line represent larger prediction errors.

Insights:

- **Accuracy**: The plot shows a generally positive linear trend, meaning that the predicted ratings align reasonably well with the actual ratings, though not perfectly.
- **Deviation**: There are visible deviations from the red line, meaning the model's predictions are not always accurate, but the deviations don't appear too extreme.
- **Rating Distribution**: Both predicted and actual ratings are skewed toward higher values (closer to 4 and 5), which might indicate that most movies are rated favorably by users in the dataset, or that the recommendation system is biased toward predicting higher ratings.

Discussion

Insights from Model Predictions

The results show that the model can accurately predict user ratings based on a few critical features:

- **Years Since Release**: Older movies tend to receive lower ratings.
- **User Average Rating**: Users who rate more movies tend to provide consistent scores.
- **Movie Average Rating**: Highly rated movies are generally predicted more accurately by the model.

These insights can be used to enhance movie recommendation systems, improving personalized suggestions for users based on their preferences and rating history.

Model Limitations

Despite the strong performance, there are some limitations:

- **Sparsity of Data**: Some users have rated very few movies, making it difficult for the model to generalize well for these users.
- **Feature Limitations**: Additional features such as movie genres or user demographics could potentially improve model performance.

Conclusion

In conclusion, we have successfully built a predictive model that forecasts movie ratings with high accuracy. The model demonstrated a strong performance with a low RMSE of r round(rmse, 4). The insights drawn from this analysis can be used to enhance movie recommendation systems, providing users with better-tailored suggestions based on their historical ratings.

Recommendations for Future Work

Model Improvements

- **Feature Engineering**: Incorporating additional features like movie genres, user demographics, or interaction effects between users and movies could further improve prediction accuracy.
- **Algorithm Comparison**: Future work could compare the performance of the Random Forest model with other machine learning models, such as Gradient Boosting Machines or Neural Networks.
- **Cross-Validation**: Implementing more advanced cross-validation techniques to optimize hyperparameters could lead to further improvements in model performance.

Business Application

This model could be integrated into a **real-time recommendation engine** for a streaming platform. By continuously updating the model with new user ratings, the system could improve user engagement and satisfaction by providing highly personalized movie suggestions.