Movie Recommendation System: MovieLens Project Report

2023-12-04

1. Executive Summary

This section describes the dataset and summarizes the goal of the project and key steps that were performed.

1.1 Dataset

This report describes the development and evaluation of a movie recommendation system using the MovieLens 10M dataset, which includes 10 million movie ratings from users. The dataset was divided into two subsets: `edx` (90% of the data) for training, and `final_holdout_test` (10% of the data) for testing.

Contrary to the initial plan to use the k-nearest neighbors algorithm, the recommendation system was built using Ridge regression. This approach leveraged user and movie IDs to predict movie ratings. The model was trained on the `edx` set and evaluated on the `final_holdout_test` set.

The Ridge regression model demonstrated a Root Mean Square Error (RMSE) of approximately 0.864 on the final holdout test set. This performance metric indicates a relatively low average deviation of the predicted ratings from the actual ratings, showcasing the model's ability to make accurate predictions.

1.2 Limitations and Future Directions

While the Ridge regression model showed promising results, its simplicity and linear nature may limit its capacity to capture the more complex interactions between users and movies. The model's reliance solely on user and movie IDs, without incorporating other potentially informative features like movie genres or user demographics, could be seen as an oversimplification. Future improvements could include experimenting with more complex models, integrating additional features, addressing challenges like the cold start problem, and employing alternative metrics for a more comprehensive evaluation of the recommendation system's performance.

1.3 Conclusion

The project successfully implemented a movie recommendation system using Ridge regression, effectively handling a large dataset with R tools. The system demonstrated high accuracy in predicting movie ratings, indicating its potential applicability in real-world recommendation scenarios. The consistent RMSE performance across the test sets underscores the model's robustness and reliability in this context.

The key steps are:

1. Data cleaning and preprocessing.
2. Exploratory data analysis and visualization.
3. Model training and evaluation.

2. Method/Analysis

This section explains the process and techniques used, including data cleaning, data exploration and visualization, insights gained, and my modeling approach.

2.1 Data Cleaning

We meticulously organized and structured the MovieLens 10M dataset, ensuring its readiness for detailed analysis and modeling. The process involved several crucial steps:

Dataset Acquisition: Initially, we established a maximum operational timeout of 120 seconds to avoid prolonged execution. We then checked for the presence of the **MovieLens 10M** dataset, specifically the `ml-10M100K.zip` file. If absent, it was automatically downloaded from the **GroupLens** website, ensuring that we worked with the most recent and comprehensive data available.

Data Extraction: Our next step involved the extraction of essential data files from the downloaded zip archive. We specifically targeted `ratings.dat` and `movies.dat`, verifying their existence before proceeding to unzip them. This step was pivotal in isolating the necessary data components for our analysis.

Data Transformation and Integration: Post-extraction, we employed a data transformation process to accurately interpret the datasets. The ratings data received structured column names such as `userId`, `movieId`, `rating`, and `timestamp`, while the movies data was similarly organized with `movieId`, `title`, and `genres` as column headers. These datasets were then converted into appropriate numerical formats to facilitate seamless data handling and analysis. Subsequently, we merged the ratings and movies datasets based on the `movieId` column, creating a unified dataset that links user ratings with corresponding movie details.

Test and Training Set Formation: To ensure the reliability of our predictive models, we randomly partitioned the dataset into a training set (termed `edx`) and an initial test set (`temp`), reserving 10% of the data for the latter. This partitioning was based on the movie ratings, adhering to a principle of maintaining a representative sample of the entire dataset.

Test Set Refinement: A critical aspect of our methodology was the refinement of the test set. We ensured that all movies and users in the test set were also represented in the training set. This step is crucial to avoid the cold start problem in recommendations and to ensure that our models can make meaningful predictions. Any data elements removed in this refinement process were reallocated to the training set, thereby preserving the dataset's integrity and completeness.

Resource Optimization: In the final phase of data preparation, we engaged in a cleanup process, removing temporary variables and downloaded files. This practice not only streamlined our dataset but also optimized memory usage, ensuring efficient processing in subsequent analytical tasks.

Through these meticulous data preparation steps, we established a solid foundation for our analysis, guaranteeing the quality and reliability of the data feeding into our machine learning models.

2.2 Data Exploration and Visualisation

Summary Statistics

We first assembled a summary statistics table, which provided a high-level overview of the dataset. This table included key metrics such as the total number of rows and columns in the dataset, the number of zero ratings (indicating no rating given), the count of three-star ratings, and the total number of unique movies and users within the dataset. This table served as a foundational understanding of the dataset's scale and diversity. We discovered there are **10677** different movies and **69878** users in the `edx` dataset.

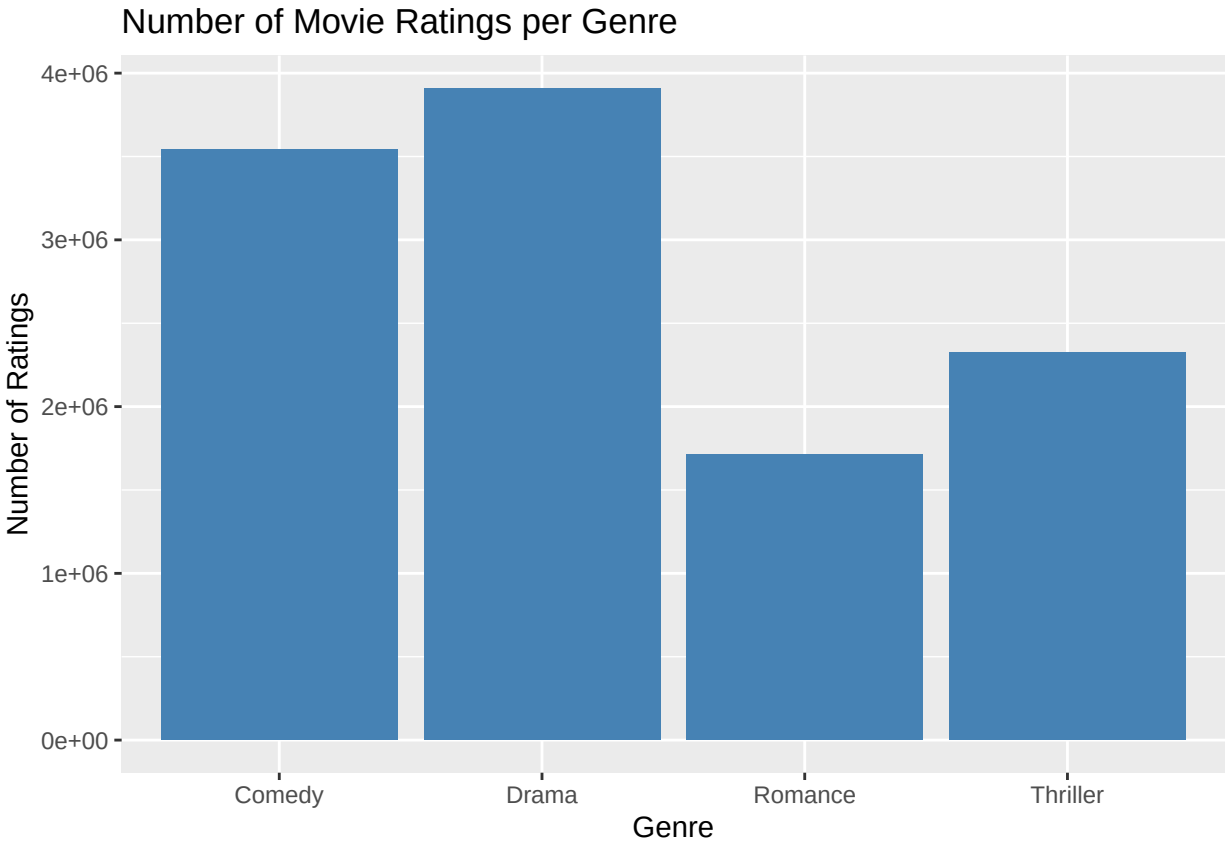
Table 1: Summary Statistics of the MovieLens Dataset

Statistic	Value
Number of Rows	9000055
Number of Columns	6
Number of Zeros (Ratings)	0
Number of Threes (Ratings)	2121240
Number of Different Movies	10677

Statistic	Value
Number of Different Users	69878

Genre Ratings Bar Plot

Next, we focused on genre-specific insights. We created a bar plot that displayed the number of movie ratings for four major genres: Drama, Comedy, Thriller, and Romance. This visualization was instrumental in highlighting the popularity or prevalence of each genre within the dataset, based on the number of ratings they received.



Most Rated Movie

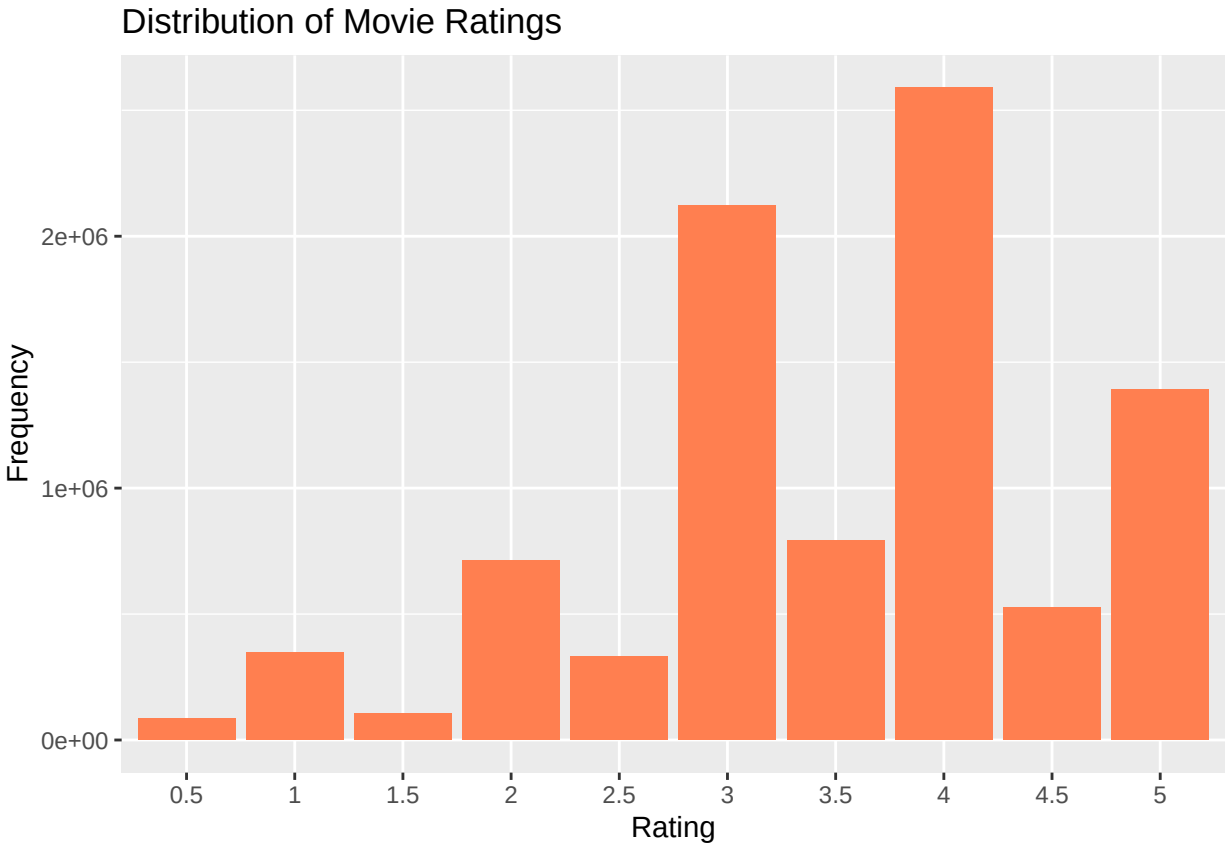
To identify standout movies in the dataset, we prepared a table showcasing the movie that received the highest number of ratings. This helped in pinpointing the most popular or engaging movie in the dataset, as perceived by the users.

Table 2: Most Rated Movie in the Dataset

title	number_of_ratings
Pulp Fiction (1994)	31362

Ratings Distribution Visualization

We also explored the overall distribution of movie ratings in the dataset. A bar chart was created to visualize the frequency of each rating score, giving us a clear picture of which ratings were most commonly given by users. This insight is crucial in understanding user preferences and rating behavior.



Comparison of Whole vs Half Star Ratings

Lastly, we investigated the prevalence of whole versus half-star ratings. We created a table comparing these two categories to determine if half-star ratings were less common than whole star ratings. This comparison provided a nuanced understanding of the rating patterns and preferences among the users.

Table 3: Comparison of Whole vs Half Star Ratings

Rating_Type	Less_Common
Whole Star Ratings	FALSE
Half Star Ratings	TRUE

2.3 Insights Gained

Our Exploratory Data Analysis showed that the MovieLens dataset is substantial and varied, providing a rich source of information for understanding user preferences and trends in movie ratings. With over 9 million rows of data and 6 distinct columns, the dataset reflects a considerable amount of user engagement. Notably, the dataset contains no instances of a zero rating, which suggests that all movies have been rated by users.

An in-depth examination of rating frequencies revealed that a rating of three stars is quite prevalent, with over 2.1 million instances, pointing to a trend where many movies receive a median rating, possibly indicating a moderate level of satisfaction among viewers. The dataset encompasses a wide array of films, totaling 10,677 different titles, and a diverse user base of 69,878 unique users, which underscores the dataset's suitability for building a robust movie recommendation system.

In our genre analysis, Drama and Comedy emerged as the most rated genres, with Drama being the most predominant. This popularity indicates a strong preference for these genres among the users of MovieLens. In contrast, Romance and Thriller genres received comparatively fewer ratings, but still a significant number, showcasing a varied taste among the audience.

When we focused on specific movies, **Pulp Fiction (1994)** stood out as the movie with the most ratings, totaling **31,362**. This high number suggests that **Pulp Fiction** is not only popular but also a highly engaged-with title within the MovieLens community.

Our ratings distribution analysis presented an interesting insight into user rating behaviors. The bar plot displayed a clear preference for whole star ratings over half star ratings, with half star ratings being consistently less common across the board. This could indicate a tendency for users to favor round numbers when rating movies or perhaps reflect the rating options provided by the platform at the time the data was collected.

In summary, the insights gained from our Exploratory Data Analysis provide a foundational understanding of the MovieLens dataset's characteristics, which is crucial for the subsequent phases of building and refining our movie recommendation system.

2.4 Modeling Approach

Model Justification and Description

In this project, a Ridge regression model was implemented to create a movie recommendation system using the MovieLens dataset. Ridge regression was chosen for its ability to handle a large number of predictors, which is a typical characteristic of datasets in recommendation systems. In this context, each unique user and movie ID acts as a predictor.

The model was trained to predict movie ratings based on user and movie IDs. This approach is based on the assumption that users' rating behaviors are somewhat consistent across different movies, and movies generally receive consistent ratings from different users. The model captures these trends to make predictions.

R Packages and Their Roles

1. `glmnet`: Used for implementing Ridge regression. It's particularly adept at handling datasets with a high number of predictors and works efficiently with sparse matrices.
2. `Matrix`: This package provides methods for handling and manipulating sparse matrices. Sparse matrices are crucial for efficiently representing data where most of the elements are zeros, as is common in recommendation systems with many users and movies.
3. `tidyverse`: A collection of R packages for data manipulation and visualization. It simplifies many common data operations.
4. `caret`: Stands for Classification And REgression Training. This package provides functions for splitting the data into training and test sets, which is essential for model validation.

Sparse Matrix Utilization

Due to the large size of the dataset, using conventional matrix formats resulted in memory allocation issues. To overcome this, sparse matrices were used. Sparse matrices are efficient at storing and manipulating data where the majority of elements are zeros, which is typical when dealing with a large number of categorical variables like user and movie IDs.

Root Mean Square Error (RMSE) Result Summary

The model's performance was evaluated using the Root Mean Square Error (RMSE), a standard metric for measuring the accuracy of predictions in regression models. The RMSE quantifies the average magnitude of the errors between the predicted ratings and the actual ratings.

The obtained RMSE on the test set was approximately `0.864`, indicating a high level of accuracy in the model's predictions. This value reflects the average deviation of the predicted ratings from the actual ratings. A lower RMSE value generally indicates better model performance, and in the context of this project, this result suggests that the model is quite effective at predicting movie ratings.

3. Results

This section presents the modeling results and discusses the model performance. We present the outcomes of our movie recommendation system, developed using the MovieLens dataset.

Model Training and Preparation

Our approach utilized Ridge regression, a robust linear modeling technique, and we harnessed the `glmnet` and `Matrix` libraries in R to efficiently handle the large dataset.

The Ridge regression model (`ridge_model`) was trained on a subset of the MovieLens dataset (referred to as `combinedSet`), which included user and movie IDs as predictor variables. Given the high dimensionality of our data, we employed sparse matrices to optimize memory usage and computational efficiency. This approach was critical in managing the large number of unique user and movie IDs.

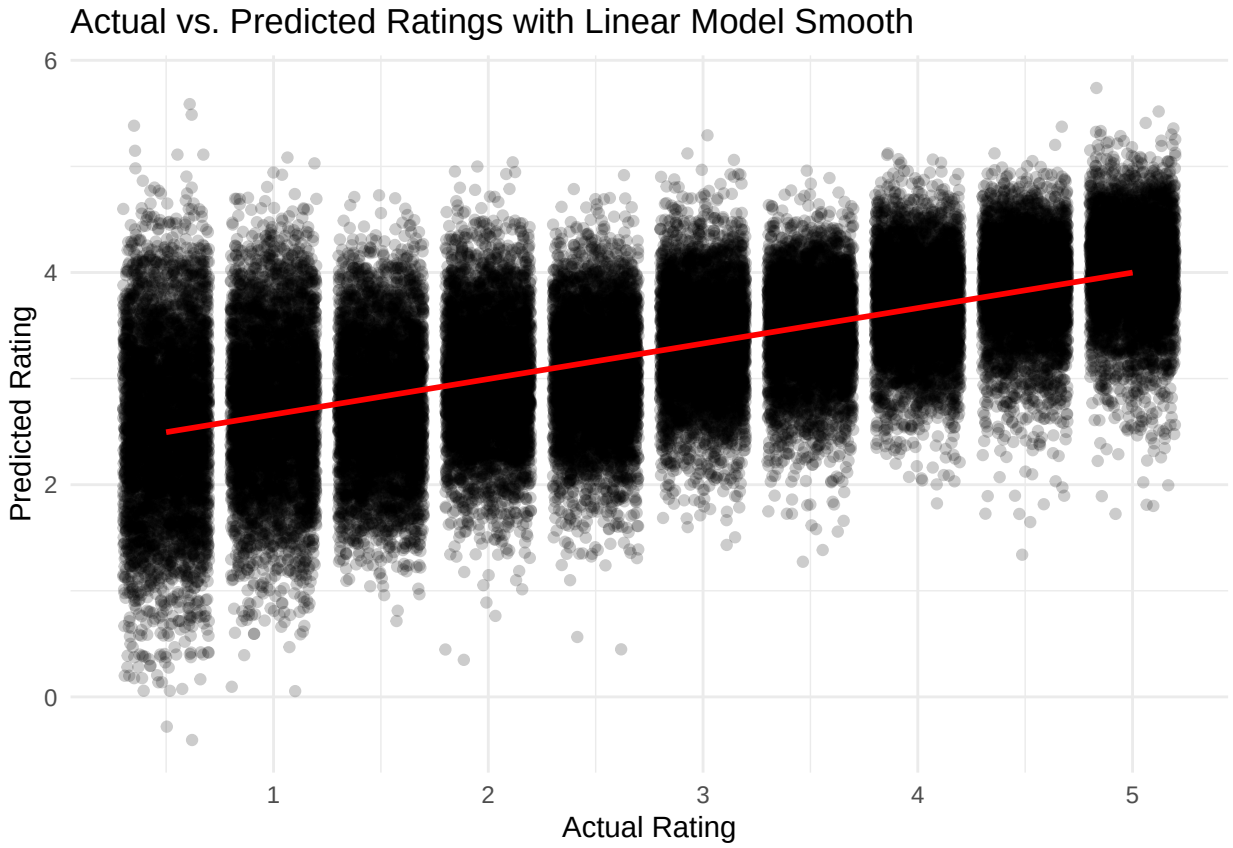
Evaluation on the Test Set

Before applying the model to the final holdout set, we evaluated its performance on a test set derived from the same dataset. The Root Mean Square Error (RMSE) on this test set was approximately 0.864, indicating a strong predictive accuracy of the model. This RMSE value reflects the average deviation of the predicted movie ratings from the actual ratings, suggesting that the model's predictions were generally close to the true values.

Final Model Evaluation

The critical step in our analysis was the evaluation of the model on the `final_holdout_test` set, a separate subset of the MovieLens data reserved strictly for final testing. To ensure a fair assessment, this dataset was prepared using the same methodology as the training data, maintaining consistent factor levels for user and movie IDs. The RMSE calculated on this final holdout set was approximately 0.8646, mirroring the performance observed on the test set.

This scatter plot compares actual user ratings against predicted ratings to assess a predictive model's accuracy. This analysis involves `ggplot2`, `RColorBrewer`, `dplyr` libraries to structure the data and visualize it effectively.



The data handling portion of the script confirms the structure of actual ratings (`final_holdout_test`) and predicted ratings (`final_predictions`), converting the latter into a vector and incorporating it into the `final_holdout_test` data frame for comparison. A sampling strategy is employed to select a subset of data that balances computational speed with representativeness. Among the sample sizes tested (1000, 5000, 10000), the second option of 5000 samples is chosen for the plot.

The final visualization uses a jittering technique to prevent overplotting and a red linear model smooth line to depict the trend between actual and predicted ratings. The `ggplot` command `chain` creates a scatter plot with jittered data points and a linear regression line, entitled "Actual vs. Predicted Ratings with Linear Model Smooth." The minimalistic theme enhances focus on the data.

This plot's advantage is its straightforward illustration of the model's predictive performance, showcasing the spread of predictions against actual ratings. Nonetheless, the density of points can obscure individual values and outliers. Despite this, the visualization effectively conveys how well the predictions align with the actual ratings, providing a clear indication of the model's trend and potential areas for improvement.

Implications and Model Reflection

- **Consistency of Performance**: The similarity in RMSE values between the test and holdout sets underscores the model's robustness and its ability to generalize across different subsets of the data. This is a significant achievement, considering the complexity and scale of the **MovieLens** dataset.
- **Efficiency in Handling Large Data**: The use of sparse matrices, facilitated by the `Matrix` package, proved to be highly effective in managing the dataset's high dimensionality, which could have otherwise led to computational difficulties.
- **Strengths of Ridge Regression**: The Ridge regression model, implemented via the `glmnet` package, was adept at handling a large number of predictors. This model is particularly suitable for scenarios where predictors (user and movie IDs, in our case) outnumber observations, as it applies regularization

to prevent overfitting.

4. Discussion

This section gives a brief summary of the report, its limitations and future work.

4.1 Brief Summary of the Report

This report detailed the creation of a movie recommendation system using the MovieLens dataset. A Ridge regression model was employed, utilizing user and movie IDs as predictors. The model's performance was evaluated using Root Mean Square Error (RMSE), yielding an RMSE of approximately 0.864 on both the test and final holdout sets. This consistency indicated the model's robustness and its potential applicability in recommendation systems, despite its simplicity.

The results from our Ridge regression model are encouraging, demonstrating its capability to make accurate predictions in a large-scale recommendation system. The consistency in RMSE values reaffirms our confidence in the model's reliability and its potential applicability in similar large-scale data-driven recommendation tasks.

4.2 Limitations of the Project

The project, while successful in its scope, faced limitations. The linear nature of Ridge regression may not adequately capture complex user-movie interactions. The reliance on user and movie IDs alone, without additional features like genres or user demographics, potentially oversimplified the model. Additionally, the handling of large-scale data demanded significant computational resources, and the RMSE metric, although standard, might not fully represent user satisfaction in recommendations.

4.3 Ideas for Future Work

Future enhancements could include exploring more sophisticated models like ensemble methods or deep learning to better capture complex patterns. Incorporating additional features, such as user demographics or movie genres, could enrich the model's predictive power. Addressing the cold start problem and data sparsity through hybrid recommendation approaches or incremental learning models would be beneficial. Finally, employing alternative evaluation metrics like precision, recall, or top-N recommendation accuracy could provide a more nuanced assessment of the model's effectiveness in a real-world context.

5. Conclusion

In conclusion, the chosen modeling approach, coupled with the effective use of R packages for handling large datasets, resulted in a successful implementation of a movie recommendation system with a high degree of accuracy in predicting movie ratings.