MovieLens Movie Recommendation System: HarvardX Data Science Capstone Project

Thant Thiha

2024-09-21

Introduction

In this project, I aim to build a movie recommendation system using the `MovieLens` 10M dataset provided by HarvardX. The goal is to predict movie ratings based on user preferences using collaborative filtering techniques. The performance of the model will be evaluated using **Root Mean Squared Error (RMSE)** to measure the accuracy of the predicted ratings compared to the actual ratings.

Key steps are 1. Data Understanding and Preparation, 2. Exploratory Data Analysis, 3. Model Development and, 4. Final Model Evaluation.

1. Data Preparation

This section describes the data preparation process, including downloading the `MovieLens` dataset, merging movie and rating information, and splitting the data into **training** `edx` (90%) and **evaluation** `final_holdout_test` (10%) sets.

2. Exploratory Data Analysis (EDA)

In this section, let's explore the data to understand the **distribution of ratings**, **the number of unique users and movies**, and other key characteristics of the dataset.

```
# Check the structure of the edx dataset
str(edx)
```

```
## 'data.frame':    9000055 obs. of  6 variables:
##  $ userId   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ movieId  : int  122 185 292 316 329 355 356 362 364 370 ...
##  $ rating   : num  5 5 5 5 5 5 5 5 5 5 ...
##  $ timestamp: int  838985046 838983525 838983421 838983392 838983392 838984474 838983653 838984885 83
##  $ title    : chr  "Boomerang (1992)" "Net, The (1995)" "Outbreak (1995)" "Stargate (1994)" ...
##  $ genres   : chr  "Comedy|Romance" "Action|Crime|Thriller" "Action|Drama|Sci-Fi|Thriller" "Action|A
```

```
#Check a summary of the dataset
summary(edx)
```

```
##      userId        movieId          rating        timestamp
##  Min.   :    1   Min.   :    1   Min.   :0.500   Min.   :7.897e+08
##  1st Qu.:18124   1st Qu.:  648   1st Qu.:3.000   1st Qu.:9.468e+08
##  Median :35738   Median : 1834   Median :4.000   Median :1.035e+09
##  Mean   :35870   Mean   : 4122   Mean   :3.512   Mean   :1.033e+09
##  3rd Qu.:53607   3rd Qu.: 3626   3rd Qu.:4.000   3rd Qu.:1.127e+09
##  Max.   :71567   Max.   :65133   Max.   :5.000   Max.   :1.231e+09
##     title              genres
```

```
##  Length:9000055     Length:9000055
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```

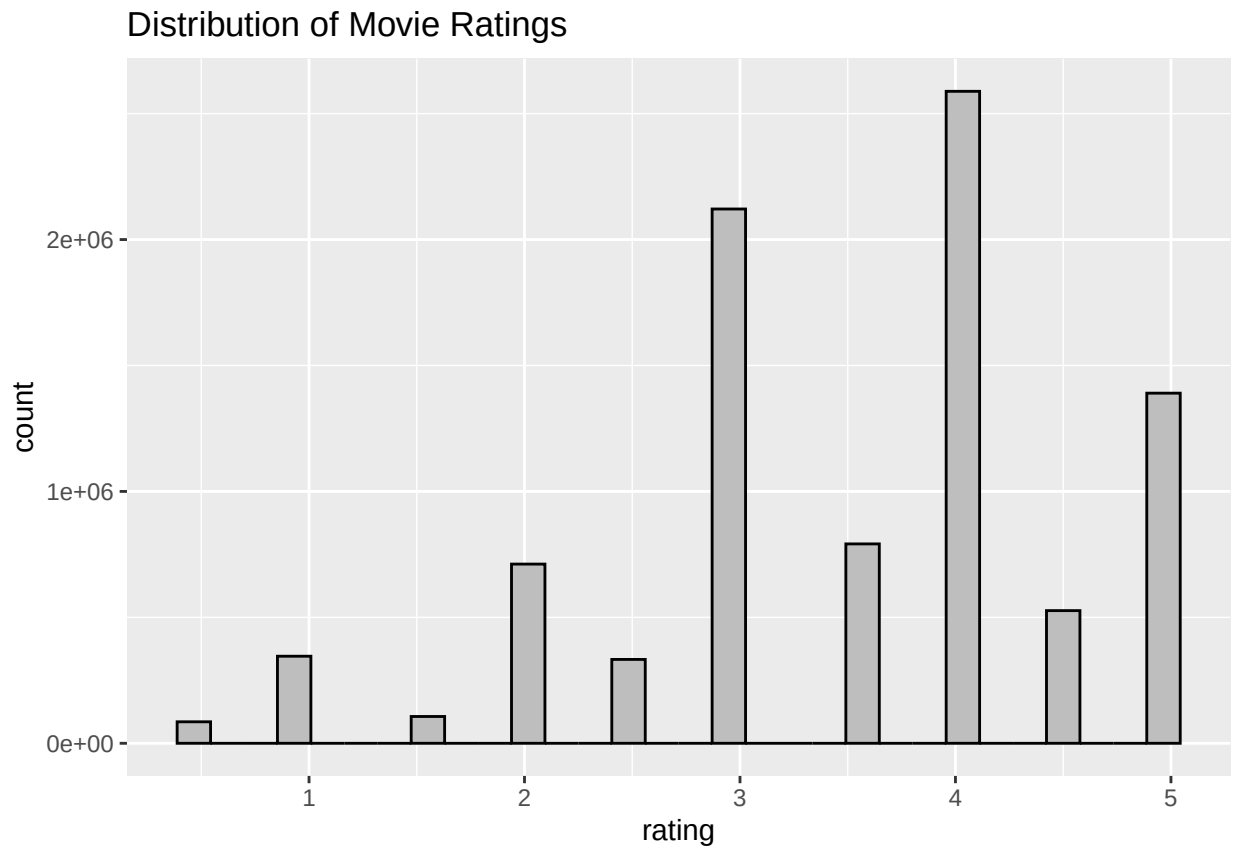
```
# Count unique users and movies
n_users <- n_distinct(edx$userId)
cat("No of unique users:", n_users, "\n")
```

```
## No of unique users: 69878
```

```
n_movies <- n_distinct(edx$movieId)
cat("No of unique movies:", n_movies, "\n")
```

```
## No of unique movies: 10677
```

```
# Distribution of ratings
edx %>%
  ggplot(aes(rating)) +
  geom_histogram(bins = 30, color = "black", fill = "grey") +
  ggtitle("Distribution of Movie Ratings")
```



3. Model Development

This section involves building multiple models to progressively improve the prediction accuracy. We will start with a baseline model and incrementally add complexity with regularization by using movie and user effects.

3.1. Baseline Model - Global Average

This simple model predicts the average rating for all movies.

```
# Calculate the global average rating
mu <- mean(edx$rating)

# Calculate RMSE for the global average model
rmse_global_avg <- sqrt(mean((edx$rating - mu)^2))

cat("RMSE Global Average Rating:", rmse_global_avg, "\n")
```

```
## RMSE Global Average Rating: 1.060331
```

3.2. Movie Effect Model

This model adjusts predictions by incorporating the specific effect of each movie.

```
# Movie Effect Model: deviation of each movie's average rating from the global average
movie_avgs <- edx %>%
  group_by(movieId) %>%
  summarize(b_i = mean(rating - mu))

# Predict ratings using movie effects
validation_preds <- edx %>%
  left_join(movie_avgs, by = "movieId") %>%
  mutate(pred = mu + b_i) %>%
  pull(pred)

# RMSE calculation
rmse_movie_effect <- RMSE(edx$rating, validation_preds)
cat("RMSE of Movie Effect Model:", rmse_movie_effect, "\n")
```

```
## RMSE of Movie Effect Model: 0.9423475
```

3.3. Regularized User and Movie Effect Model

To avoid overfitting, we apply regularization to movie and user effects.

```
# Define a regularization parameter
lambda <- 5

# Regularize movie effects
movie_avgs_reg <- edx %>%
  group_by(movieId) %>%
  summarize(b_i = sum(rating - mu) / (n() + lambda))

# Regularize user effects
user_avgs <- edx %>%
  left_join(movie_avgs_reg, by = "movieId") %>%
  group_by(userId) %>%
  summarize(b_u = sum(rating - mu - b_i) / (n() + lambda))

# Predict ratings with regularized movie and user effects
validation_preds <- edx %>%
  left_join(movie_avgs_reg, by = "movieId") %>%
  left_join(user_avgs, by = "userId") %>%
```

```
  mutate(pred = mu + b_i + b_u) %>%
  pull(pred)

# Calculate RMSE for the Movie and User effect model
rmse_final_model <- RMSE(edx$rating, validation_preds)
cat("RMSE of Regularized Movie + User Effect Model:", rmse_final_model, "\n")
```

```
## RMSE of Regularized Movie + User Effect Model: 0.8570452
```

4. Final Model Evaluation

Evaluate the selected model on the `final_holdout_test` set to obtain the final RMSE.

```
# Predict on the final_holdout_test set using the best model
final_predictions <- final_holdout_test %>%
  left_join(movie_avgs_reg, by = "movieId") %>%
  left_join(user_avgs, by = "userId") %>%
  mutate(pred = mu + b_i + b_u) %>%
  pull(pred)

# Calculate RMSE on the final holdout test set
rmse_holdout <- RMSE(final_holdout_test$rating, final_predictions)
cat("Final RMSE on Holdout Test Set:", rmse_holdout, "\n")
```

```
## Final RMSE on Holdout Test Set: 0.8648177
```

Summary of Results

The analysis of the `MovieLens` dataset revealed an RMSE of global average rating of **1.0603**. The Movie Effect Model achieved an RMSE of **0.9423**, while the Regularized Movie + User Effect Model improved accuracy with an RMSE of **0.8570**. The final model's RMSE on the holdout test set was **0.8648**, indicating good predictive performance.

Conclusion

This report highlights the importance of considering both movie and user effects in predicting movie ratings. Despite achieving solid results, the analysis has limitations, including reliance on historical data and potential outlier influence. Future work should explore advanced techniques and incorporate additional features to enhance model performance and insights.