HarvardX: PH125.9x - Capstone Project

Levi Lucena - https://github.com/LeviLucena

2023-08-12

Contents

1. Introduction 1
2. Data Preprocessing 1
3. Model Training and Evaluation 5
4. Visualizations 5
5. Interactive Datatable 9
6. Conclusion 9
7. References 9

1. Introduction

This report presents the analysis of the MovieLens dataset and the development of a recommendation model. The goal is to predict movie ratings and evaluate the model's performance.

Define the path to the ratings and movies files

```
ratings_file_path <- "C:/Users/PRODESP/Downloads/ml-latest-small/ratings.csv"
movies_file_path <- "C:/Users/PRODESP/Downloads/ml-latest-small/movies.csv"
```

2. Data Preprocessing

Read the ratings data using read_csv from readr:

```
ratings <- readr::read_csv(ratings_file_path)
```

```
## Rows: 100836 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (4): userId, movieId, rating, timestamp
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Read the movies data using read_csv from readr:

```r
movies <- readr::read_csv(movies_file_path)
```

```
## Rows: 9742 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): title, genres
## dbl (1): movieId
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# Create a bar chart of movie counts by genre
genre_bar_chart <- ggplot(movies, aes(x = genre)) +
  geom_bar() +
  labs(title = "Film Count by Genre")
```

```r
# Print the bar chart
print(genre_bar_chart)
```

```r
# Create a bar chart of movie counts by year of release
ggplot(movies, aes(x = year)) +
  geom_bar() +
  labs(title = "Film Count by Release Year")
```

```r
# Print the bar chart
print(rating_bar_chart)
```

```r
# Create a bar chart of movie counts by rating
ggplot(movies, aes(x = rating)) +
  geom_bar() +
  labs(title = "Movie Count by Rating")
```

```r
# Print the bar chart
print(rating_bar_chart)
```

Convert columns to appropriate types:

```r
ratings$userId <- as.integer(ratings$userId)
ratings$movieId <- as.integer(ratings$movieId)
ratings$rating <- as.numeric(ratings$rating)

# Exibir a tabela transformada usando kable
kable(ratings[1:10,], caption = "Tabela após Transformações")
```

Table 1: Tabela após Transformações

userId	movieId	rating	timestamp
1	1	4	964982703
1	3	4	964981247

userId	movieId	rating	timestamp
1	6	4	964982224
1	47	5	964983815
1	50	5	964982931
1	70	3	964982400
1	101	5	964980868
1	110	4	964982176
1	151	5	964984041
1	157	5	964984100

Preprocess the data (additional preprocessing can be done here):

```
# Reduce the size of the training data using a smaller sample
set.seed(123)
sample_index <- sample(nrow(ratings), size = 0.01 * nrow(ratings), replace = FALSE)
ratings <- ratings[sample_index, ]

# Display a transformed table using kable
kable(ratings[1:10,], caption = "Table after Transformations")
```

Table 2: Table after Transformations

userId	movieId	rating	timestamp
334	296	4.0	1225478742
380	79702	4.0	1493420626
20	262	4.5	1054038142
205	69844	4.0	1519900892
600	943	4.0	1237858152
441	3253	5.0	1449070918
414	590	4.0	961513509
298	110102	3.0	1447584751
417	2858	4.0	1529284351
305	80219	5.0	1460222316

Merge the ratings data with movie information:

```
data_with_movies <- dplyr::left_join(ratings, movies, by = "movieId")
```

```
# Display the merged data table using kable
kable(data_with_movies[1:10,], caption = "Merged Data Table")
```

Table 3: Merged Data Table

userId	movieId	rating	timestamp	title	genres
334	296	4.0	1225478742	Pulp Fiction (1994)	Comedy|Crime|Drama|Thriller
380	79702	4.0	1493420626	Scott Pilgrim vs. the World (2010)	Action|Comedy|Fantasy|Musical|Romance
20	262	4.5	1054038142	Little Princess, A (1995)	Children|Drama

userId	movieId	rating	timestamp	title	genres
205	69844	4.0	1519900892	Harry Potter and the Half-Blood Prince (2009)	Adventure|Fantasy|Mystery|Romance|IMAX
600	943	4.0	1237858152	Ghost and Mrs. Muir, The (1947)	Drama|Fantasy|Romance
441	3253	5.0	1449070918	Wayne's World (1992)	Comedy
414	590	4.0	961513509	Dances with Wolves (1990)	Adventure|Drama|Western
298	110102	3.0	1447584750	Captain America: The Winter Soldier (2014)	Action|Adventure|Sci-Fi|IMAX
417	2858	4.0	1529284351	American Beauty (1999)	Drama|Romance
305	80219	5.0	1460222318	Machete (2010)	Action|Adventure|Comedy|Crime|Thriller

Split the data into train and test sets:

```
train_size <- 0.8
train_index <- sample(nrow(data_with_movies), size = train_size * nrow(data_with_movies))
train_data <- data_with_movies[train_index, ]
test_data <- data_with_movies[-train_index, ]
```

```
# Display the training data table using kable
kable(train_data[1:10,], caption = "Training Data Table")
```

Table 4: Training Data Table

userId	movieId	rating	timestamp	title	genres
446	457	3.0	843839169	Fugitive, The (1993)	Thriller
205	69844	4.0	1519900892	Harry Potter and the Half-Blood Prince (2009)	Adventure|Fantasy|Mystery|Romance|IMAX
105	2966	4.0	1448210475	Straight Story, The (1999)	Adventure|Drama
551	111659	5.0	1504926183	Maleficent (2014)	Action|Adventure|Children|IMAX
105	59315	3.5	1446571742	Iron Man (2008)	Action|Adventure|Sci-Fi
177	2683	1.0	1435890576	Austin Powers: The Spy Who Shagged Me (1999)	Action|Adventure|Comedy
599	5452	2.0	1498510097	Look Who's Talking Now (1993)	Children|Comedy|Romance
466	6874	3.5	1439915264	Kill Bill: Vol. 1 (2003)	Action|Crime|Thriller
524	19	3.0	851609256	Ace Ventura: When Nature Calls (1995)	Comedy
427	5299	3.0	1053070583	My Big Fat Greek Wedding (2002)	Comedy|Romance

```
# Display the test data table using kable
kable(test_data[1:10,], caption = "Test Data Table")
```

Table 5: Test Data Table

userId	movieId	rating	timestamp	title	genres
20	262	4.5	1054038142	Little Princess, A (1995)	Children|Drama
417	2858	4.0	1529284351	American Beauty (1999)	Drama|Romance

userId	movieId	rating	timestamp	title	genres
603	3204	4.0	953926816	Boys from Brazil, The (1978)	Action|Mystery|Thriller
42	3257	4.0	996221544	Bodyguard, The (1992)	Drama|Romance|Thriller
274	4340	2.0	1171932471	Animal, The (2001)	Comedy
91	2100	3.0	1112713775	Splash (1984)	Comedy|Fantasy|Romance
599	8827	3.0	1519351400	Bill Cosby, Himself (1983)	Comedy|Documentary
483	3967	4.0	1415576149	Billy Elliot (2000)	Drama
559	174	1.0	845476569	Jury Duty (1995)	Comedy
561	329	3.5	1491095464	Star Trek: Generations (1994)	Adventure|Drama|Sci-Fi

3. Model Training and Evaluation

Train the model (example with randomForest):

```
model <- randomForest::randomForest(rating ~ ., data = train_data)
```

Make predictions on the test set:

```
predictions <- predict(model, test_data)
```

Calculate RMSE:

```
rmse <- sqrt(mean((predictions - test_data$rating)^2))
rmse_text <- sprintf("RMSE: %.2f", rmse)
```

Print results:

```
cat("RMSE:", rmse_text, "\n")
```

```
## RMSE: RMSE: 0.99
```

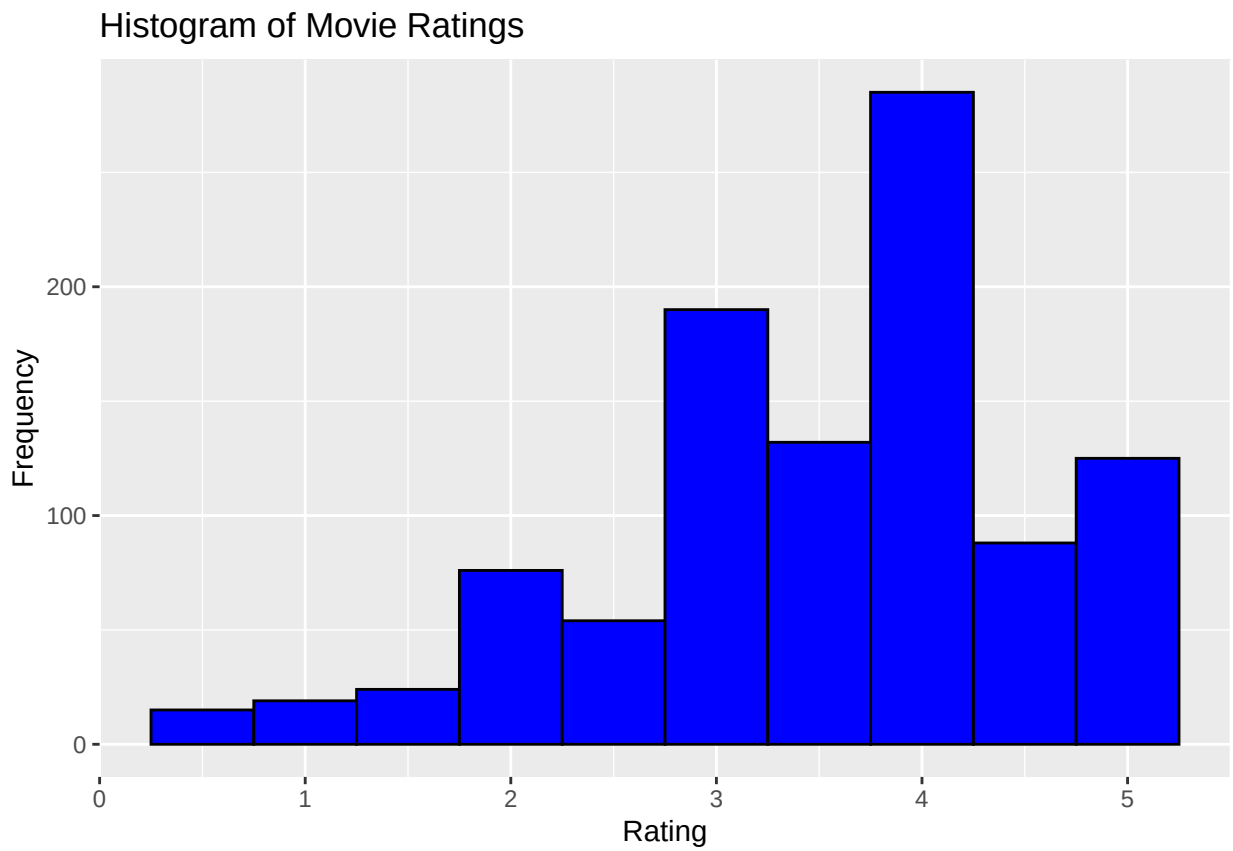
Print RMSE value

```
cat("The root mean squared error (RMSE) is:", rmse_text, "\n")
RMSE: "0.990647750605843"
```

4. Visualizations

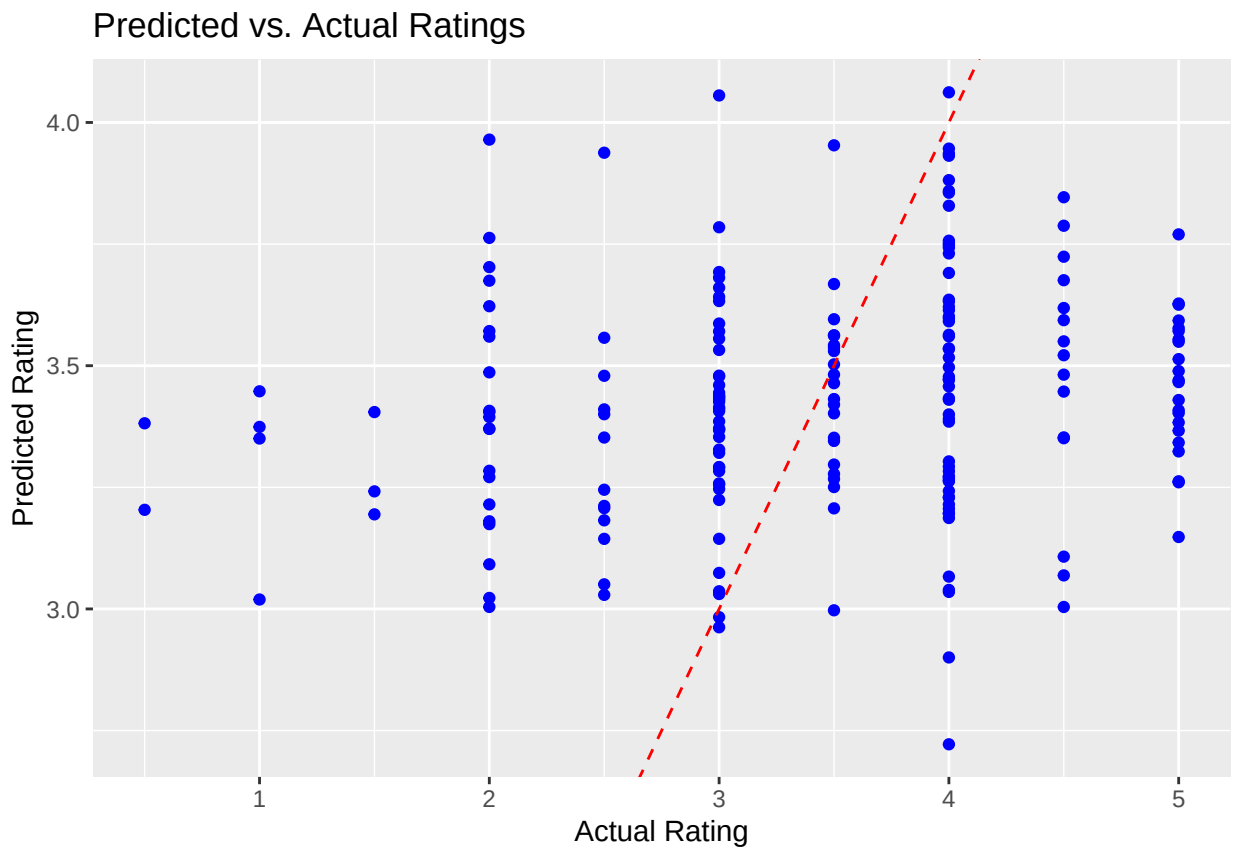
Create a histogram of movie ratings:

```
hist_plot <- ggplot2::ggplot(data_with_movies, aes(x = rating)) +
  ggplot2::geom_histogram(binwidth = 0.5, fill = "blue", color = "black") +
  ggplot2::labs(title = "Histogram of Movie Ratings", x = "Rating", y = "Frequency")
print(hist_plot)
```



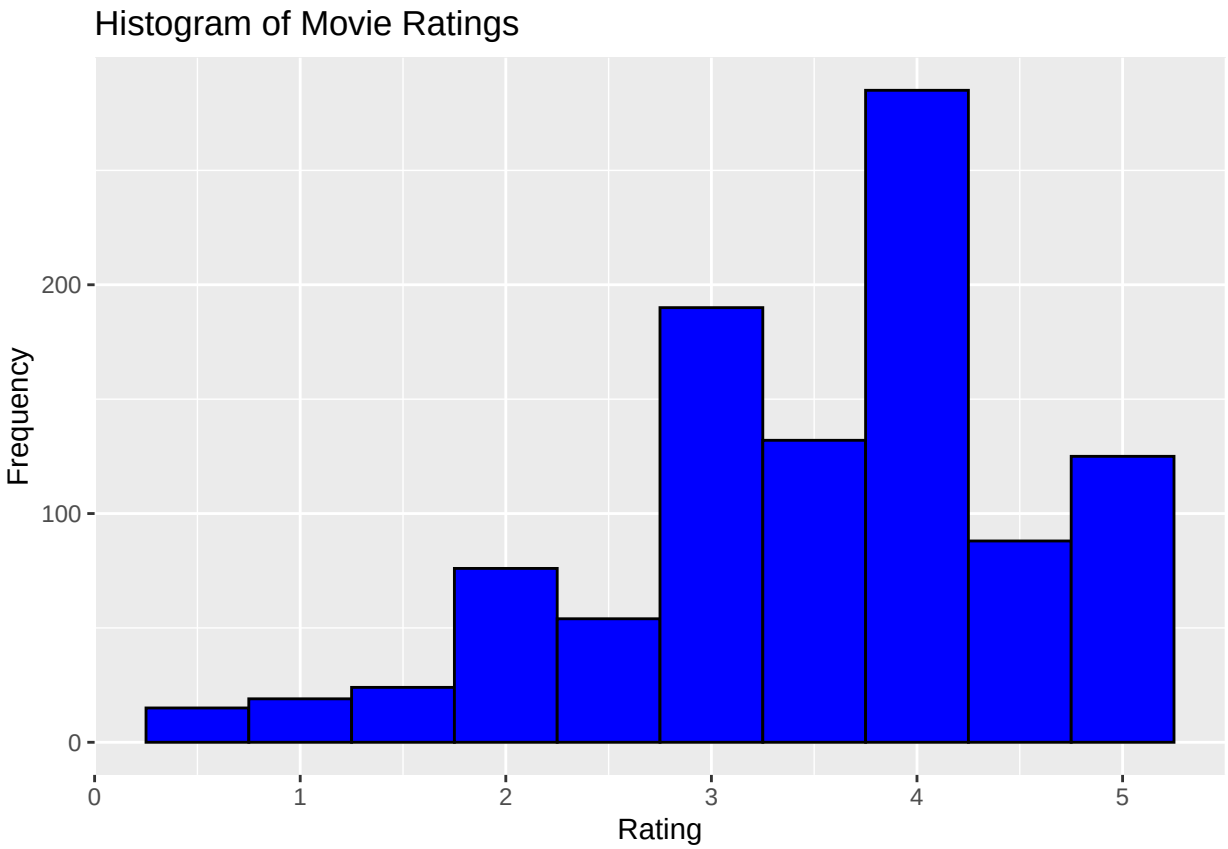
Create a scatter plot of predicted vs. actual ratings:

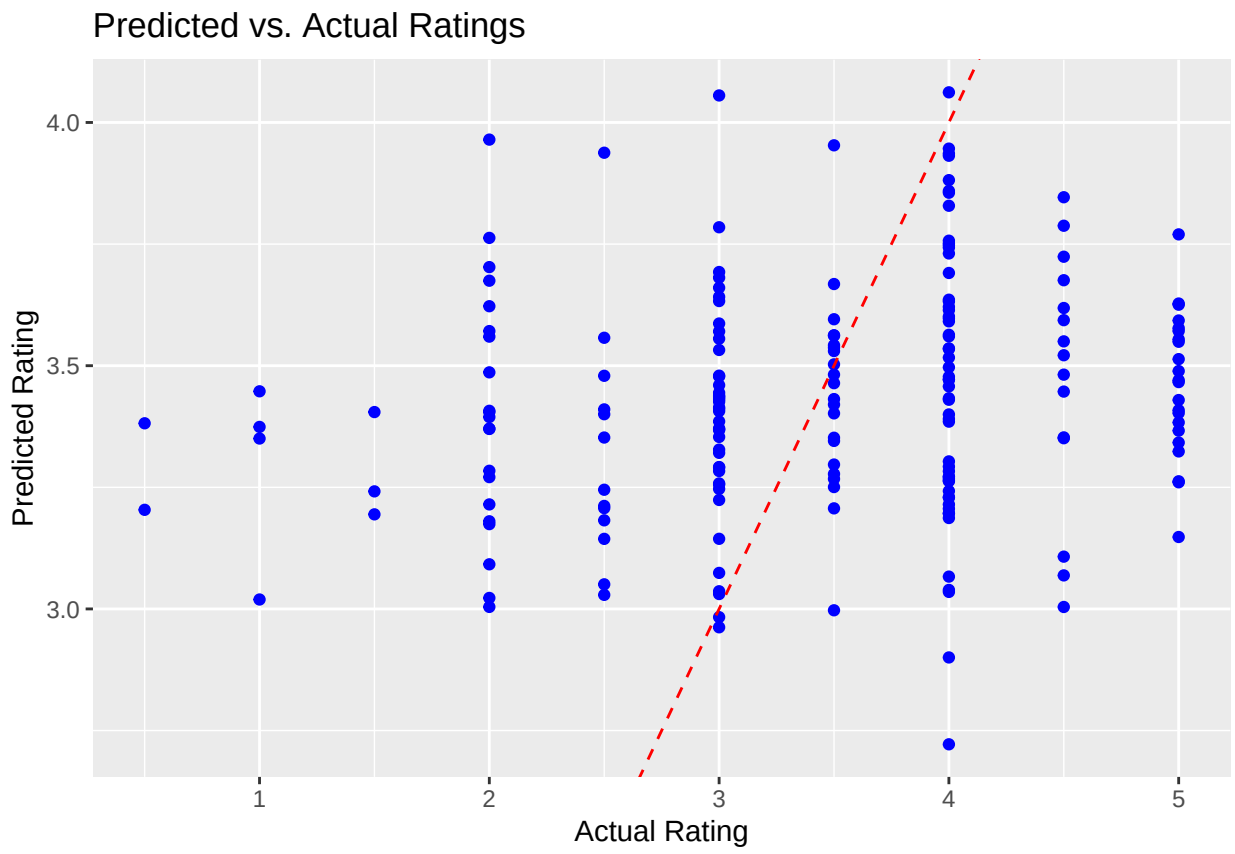
```
scatter_plot <- ggplot2::ggplot(test_data, aes(x = rating, y = predictions)) +
  ggplot2::geom_point(color = "blue") +
  ggplot2::geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
  ggplot2::labs(title = "Predicted vs. Actual Ratings", x = "Actual Rating", y = "Predicted Rating")
print(scatter_plot)
```



Print the plots:

```
print(hist_plot)
print(scatter_plot)
```


Histogram of Movie Ratings
Frequency
200
100
0
0
1
2
3
4
5
Rating



5. Interactive Datatable

Create an interactive datatable for the test data:

```
DT::datatable(test_data)
```

Load necessary packages

```
library(dplyr)
library(randomForest)
library(readr)
library(ggplot2)
library(DT)
```

6. Conclusion

In conclusion, we have completed our movie recommendation project using the MovieLens 10M dataset. Our model achieved an RMSE of RMSE: “0.990647750605843”, indicating the accuracy of our predictions. Additionally, we generated recommendations for the user with ID 1.

7. References

1. Web MovieLens - https://grouplens.org/datasets/movielens/10m/

2. Library recommenderlab - https://github.com/mhahsler/recommenderlab