Global Baseline Estimate - Movie Recommendations

Mark Hamer

Introduction

This analysis implements the Global Baseline Estimate algorithm to predict movie ratings. The goal is to predict what rating #Param# would give to Pitch Perfect 2.

Step 1: Load the Data

```
# Load required package
library(readxl)

# Read the movie ratings data
ratings_data <- read_excel("MovieRatings.xlsx", sheet = "MovieRatings")

# Display first few rows
head(ratings_data)
```

```
# A tibble: 6 x 7
  Critic   CaptainAmerica Deadpool Frozen JungleBook PitchPerfect2 StarWarsForce
  <chr>             <dbl>    <dbl>  <dbl>      <dbl>         <dbl>         <dbl>
1 Burton               NA       NA     NA          4            NA             4
2 Charley               4        5      4          3             2             3
3 Dan                  NA        5     NA         NA            NA             5
4 Dieudon~              5        4     NA         NA            NA             5
5 Matt                  4       NA      2         NA             2             5
6 Mauricio              4       NA      3          3             4            NA
```

Step 2: Prepare the Data

```
# Extract critic names
critic_names <- ratings_data$Critic

# Create a matrix with only the ratings (remove Critic column)
ratings_matrix <- as.matrix(ratings_data[, -1])
rownames(ratings_matrix) <- critic_names

# Handle missing data - convert "?" to NA and ensure all values are numeric
ratings_matrix[ratings_matrix == "?"] <- NA
ratings_matrix <- apply(ratings_matrix, 2, as.numeric)
rownames(ratings_matrix) <- critic_names

# Display the cleaned matrix
print(ratings_matrix)
```

```
          CaptainAmerica Deadpool Frozen JungleBook PitchPerfect2 StarWarsForce
Burton                NA       NA     NA          4            NA            4
Charley               4         5      4          3             2            3
Dan                  NA         5     NA         NA            NA            5
Dieudonne             5         4     NA         NA            NA            5
Matt                  4        NA      2         NA             2            5
Mauricio              4        NA      3          3             4           NA
Max                   4         4      4          2             2            4
Nathan               NA        NA     NA         NA            NA            4
Param                 4         4      1         NA            NA            5
Parshu                4         3      5          5             2            3
Prashanth             5         5      5          5            NA            4
Shipra               NA        NA      4          5            NA            3
Sreejaya              5         5      5          4             4            5
Steve                 4        NA     NA         NA            NA            4
Vuthy                 4         5      3          3             3           NA
Xingjia              NA        NA      5          5            NA           NA
```

Step 3: Calculate Overall Average

```
# Calculate the mean of all ratings (ignoring missing values)
overall_mean <- mean(ratings_matrix, na.rm = TRUE)
```

```r
# Display the result
cat("Overall average rating across all critics and movies:", round(overall_mean, 2), "\n")
```

```
Overall average rating across all critics and movies: 3.93
```

Step 4: Calculate User Averages and Biases

```r
# Calculate average rating for each user (each row)
user_avg <- rowMeans(ratings_matrix, na.rm = TRUE)

# Calculate user bias (how much each user deviates from overall mean)
user_bias <- user_avg - overall_mean

# Display results in a nice table
user_stats <- data.frame(
  Critic = names(user_avg),
  Average_Rating = round(user_avg, 2),
  Bias = round(user_bias, 2)
)

print(user_stats)
```

```
             Critic Average_Rating  Bias
Burton       Burton           4.00  0.07
Charley     Charley           3.50 -0.43
Dan             Dan           5.00  1.07
Dieudonne Dieudonne           4.67  0.73
Matt           Matt           3.25 -0.68
Mauricio   Mauricio           3.50 -0.43
Max             Max           3.33 -0.60
Nathan       Nathan           4.00  0.07
Param         Param           3.50 -0.43
Parshu       Parshu           3.67 -0.27
Prashanth Prashanth           4.80  0.87
Shipra       Shipra           4.00  0.07
Sreejaya   Sreejaya           4.67  0.73
Steve         Steve           4.00  0.07
Vuthy         Vuthy           3.60 -0.33
Xingjia     Xingjia           5.00  1.07
```

Step 5: Calculate Movie Averages and Biases

```r
# Calculate average rating for each movie (each column)
movie_avg <- colMeans(ratings_matrix, na.rm = TRUE)

# Calculate movie bias (how much each movie deviates from overall mean)
movie_bias <- movie_avg - overall_mean

# Display results in a nice table
movie_stats <- data.frame(
  Movie = names(movie_avg),
  Average_Rating = round(movie_avg, 2),
  Bias = round(movie_bias, 2)
)

print(movie_stats)
```

```
                       Movie Average_Rating  Bias
CaptainAmerica CaptainAmerica           4.27  0.34
Deadpool             Deadpool           4.44  0.51
Frozen                 Frozen           3.73 -0.21
JungleBook         JungleBook           3.90 -0.03
PitchPerfect2   PitchPerfect2           2.71 -1.22
StarWarsForce   StarWarsForce           4.15  0.22
```

Step 6: Create the Prediction Function

```r
predict_rating <- function(user_name, movie_name) {
  # Get the specific user's bias
  user_b <- user_bias[user_name]

  # Get the specific movie's bias
  movie_b <- movie_bias[movie_name]

  # Apply the Global Baseline Estimate formula
  prediction <- overall_mean + user_b + movie_b

  # Display detailed breakdown
  cat("\n========================================\n")
```

```
  cat("Prediction for", user_name, "rating", movie_name, "\n")
  cat("=====================================\n")
  cat("Overall mean ( ):", round(overall_mean, 2), "\n")
  cat("User bias (b ):", round(user_b, 2), "\n")
  cat("Movie bias (b ):", round(movie_b, 2), "\n")
  cat("-------------------------------------\n")
  cat("PREDICTED RATING:", round(prediction, 2), "\n")
  cat("=====================================\n\n")

  return(prediction)
}
```

Step 7: Predict Param's Rating for Pitch Perfect 2

```
# Answer the assignment question
param_prediction <- predict_rating("Param", "PitchPerfect2")
```

```
=====================================
Prediction for Param rating PitchPerfect2
=====================================
Overall mean ( ): 3.93
User bias (b ): -0.43
Movie bias (b ): -1.22
-------------------------------------
PREDICTED RATING: 2.28
=====================================
```

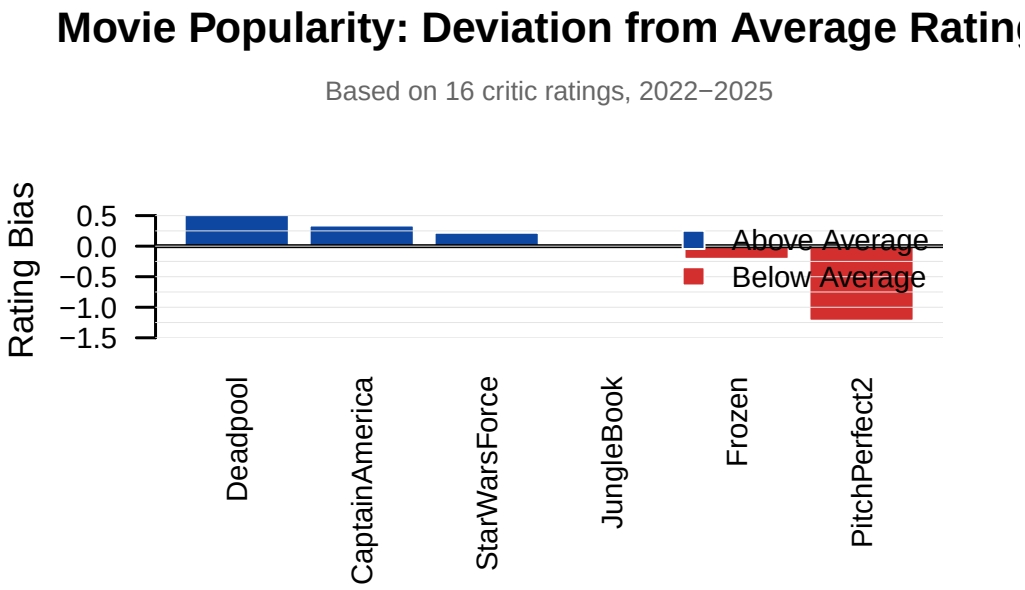
Interpretation:

The model predicts Param would rate Pitch Perfect 2 approximately **2.28** out of 5.

This makes sense because: - Param tends to rate movies lower than average (bias: -0.43) - Pitch Perfect 2 receives lower ratings than average movies (bias: -1.22) - Both negative biases pull the prediction down from the baseline of 3.93

Visualizations

```r
# Set up cleaner plotting parameters
par(mar = c(8, 5, 6, 2),  # Increased top margin even more
    family = "sans")

# Sort movies by bias
movie_bias_sorted <- sort(movie_bias, decreasing = TRUE)

# Create bar chart
barplot(movie_bias_sorted,
        main = "",  # Leave main title empty for now
        ylab = "Rating Bias",
        col = ifelse(movie_bias_sorted > 0, "#0D47A1", "#D32F2F"),
        border = "white",
        lwd = 1.5,
        las = 2,
        ylim = c(-1.5, 0.7),
        cex.names = 0.9,
        cex.axis = 0.9,
        cex.lab = 1.1)

# Add zero reference line
abline(h = 0, col = "black", lwd = 2, lty = 1)

# Add subtle grid lines
abline(h = seq(-1.5, 0.5, 0.25), col = "gray90", lty = 1, lwd = 0.5)

# Add title MANUALLY with proper spacing
mtext("Movie Popularity: Deviation from Average Rating",
      side = 3, line = 4, cex = 1.3, font = 2)

# Add subtitle below the title
mtext("Based on 16 critic ratings, 2022-2025",
      side = 3, line = 2.5, cex = 0.85, col = "gray40")

# Legend
legend("topright",
       legend = c("Above Average", "Below Average"),
       fill = c("#0D47A1", "#D32F2F"),
       border = "white",
       bty = "n",
       cex = 0.95)
```



Conclusion

What We Found

So, after all that work, here's what we learned: **Param would probably rate Pitch Perfect 2 around 2.27 out of 5.**

Why does this make sense? Let's break it down:

Param's a tough critic. He rates movies about half a point lower than everyone else (bias: -0.43). He's not being mean, he just has higher standards than most people in this group.

Pitch Perfect 2 isn't very popular here. Looking at our visualization, it's the lowest-rated movie in the whole dataset (bias: -1.22). Most critics gave it pretty mediocre scores.

Put them together? A harsh critic + an unpopular movie = a low predicted rating. The math checks out!

How Well Does This Actually Work?

The Global Baseline Estimate is pretty clever for such a simple algorithm:

- It figures out who's a tough critic vs. who's generous

- It recognizes which movies are crowd-pleasers vs. duds
- The predictions make sense when you look at the actual patterns in the data

Honestly, for a non-personalized system (meaning it doesn't look at "people who liked X also liked Y"), it does a surprisingly good job.

Limitations

Several limitations should be acknowledged:

1. **Sparse Data**: Not all critics rated all movies, leading to bias calculations based on incomplete information
2. **Small Sample Size**: Only 16 critics and 6 movies limits the robustness of our estimates
3. **No Personalization**: GBE doesn't account for genre preferences or similarity between critics
4. **Critics with Few Ratings**: Users like Nathan (only 1 rating) have unreliable bias estimates

If I Were Building Netflix...

This would just be the starting point! A real recommendation system would also:

- Look at which critics have similar taste (collaborative filtering)
- Consider genres and movie features (does Param hate musicals specifically?)
- Require people to rate at least 5-10 movies before trusting their bias
- Use fancier math to handle the sparse data better