

Alejandra Robayo, Nile Rankin-Watson, Hae Rin Hong

CS230 Data Structures

Dec 9, 2022

## Gender Imbalance in Movies

We wanted to thank our professors who taught us the concepts we needed to know and Joyce, one of the tutors, who was available for us when we had questions.

### **Introduction**

The Bechdel-Wallace Test was created in the 1980s as a curt but effective way of measuring feminism (or lack thereof) in movies. It measured a movie's feminism by determining if it had at least two named female characters and if those characters had at least one conversation that was not about a man. A scarily high number of movies failed this test, raising awareness for the lack of female representation in Hollywood. Since then, several more tests have been developed in the search of a more in-depth identifier of feminism in roles as well as behind the scenes of the movie. In this project we examined the degrees of feminism in the top fifty movies realeased in the year 2016. We examined a list of twelve individual tests that measured feminism in film using one or more of the following four categories: Behind the camera, beyond white women, female protagonists, and the supporting cast.

Using these twelve tests, we have created our own test of feminism in which we put the movies through a method we created, finding it's "feminist score". To determine the feminist score of a

movie, we picked seven of the twelve tests that we thought were good methods of measuring female equality in the film industry. Wanting to touch on all areas of the film industry we made sure to include at least one test in each of the four categories previously mentioned. The seven tests we used were the following: The Bechdel Test, the Ko Test, the Pierce Test, the Villarreal test, the Koeze-Dottle Test, the Feldman Test, and the Rees Davies Test. We developed a score ranging from 1 to 100 based on the percentage of these seven tests that were passed by the movie.

## **Method**

In our method `feministScore()` we iterated through each of the test results in a movie's 'results' vector. If the test was one of the seven mentioned, we entered an if/else statement in which a movie passing the individual test at the index gained a point in an integer counter variable we created (starting from zero). A test not part of the seven aforementioned would simply cause the method move on to the next test without performing any tasks. This allowed a total of up to seven points. We then found the percentage of tests passed out of the seven we selected by multiplying the number of points gained in the counter by 100 and dividing that number by seven to find an integer score. A movie passing all seven of the movies we selected would produce a feminist score of 100 while a movie not passing any produced a score of zero. We then returned this score as an integer. We sorted our movies in order based on their feminist scores from highest to lowest. We implemented this by placing all the movies into a priority queue and dequeuing them based on their feminist scores. In the case of a tie, we ranked the movies in

alphabetical order by title, causing a movie with a title starting higher in the alphabet to be higher on the list. For example, in the case of a tie “Arrival” came before “Bad Moms”.

## **Conclusions**

We found that when we implemented `rankMovies()`, a majority of the movies had ties. Our results show that the movie with the highest Feminist score was *Independence Day: Resurgence* with a score of 85. This is followed by the rest of the movies being categorized with scores of 71, 57, 42, 28, or 14. Movies that had scores of 71 include *Bad Moms* and *Finding Dory*, movies with a score of 57 include *Suicide Squad* and *La La Land*, movies with a score of 42 include *Moana* and *Zootopia*, movies with a score of 28 include *The Angry Birds Movie* and *The Legend of Tarzan*, and movies with a score of 14 included *Deadpool* and *Doctor Strange*. We were surprised to find that so many of the movies ended up having ties, and that movies that on the surface level would appear to have a high feminist score ended up having a lower feminist score. For example, *Moana*, a movie about a girl's journey to save her island, ended up having a lower score than *Sausage Party*, a movie about different foods coming alive. This made us realize the importance of the tests that highlight the contributions behind the screen. Though our results were surprising, we think the tests used were the best indicators of what movies had higher levels of feminism than others because they represent many distinct sectors in the film industry and accounting for the different sectors. An alternate Bechdel test we thought should be added is whether there was an actor who is a member of the LGBTQ+ community in the film, and whether they play a role that goes beyond their sexual identity. We think this would expand what our feminist Score represents, since feminism is not only concerned with women and

representation of women, but also of the LGBTQ+ community and those who do not identify with cisgender, straight males. We also think by holding this test, we are opening up to include Nonbinary people and those who don't identify with the traditional standards of gender.

### **Collaboration**

The work was divided among the three of us with an initial collaboration for the Movie and Actor Classes, and a breakout into different sectors of the project when it came to tasks 3 and 4. For the Movie and Actor class, we all met together to work on these classes, and we made it a priority to make sure everyone was contributing equally. No one specific person worked on the methods within the Movie and Actor classes, but Nile did write the javadoc and getters and setters for these classes. As for the MovieCollection, Nile and Hae Rin collaborated to work on this, and Alejandra was updated on the class since we could not all meet together every single day. We also went to office hours together so we could all get together and address what was confusing to us and so none of us were in the dark about the project. For task four, we met and brainstormed what we could need for this task. We did end up running into a roadblock here, but eventually surpassed it when we went to the second to last lecture and were introduced to priorityQueues. For the report, Nile and Alejandra worked on different aspects of it, with Nile contributing to the introduction and methods of the report and Alejandra focusing on reporting the results in the conclusion.