PLATO'S MOVIE MATRIX

Daniel Levine, Joanna Zhang

ABSTRACT

Customer reviews are meant to provide feedback for a product, often with the intention of improving equality in the public exposure and of shifting demand towards the lesser known products. However in reality, customer reviews may introduce greater inequality and unfairness. This viewpoint is proved by data analysis of the Amazon movie review dataset, which provides evidence that users' attention was reinforced by already popular movie reviews, hence widening the gap of inequality. Prior ratings of a movie anchored other users' opinions, and reviews that don't necessarily reflect the true collective sentiment of a movie.

DATASET

The Amazon movie reviews span a period of more than 10 years, from August 1997 to October 2012. There are 7,911,684 reviews in the database. Reviews include product ID, user ID, user name, helpfulness of the review determined by other users, the score the user gave to the movie, the time of the review, the summary/title of the review, and the text of the review. We used the first 10% of the reviews for our analysis.

ANALYSIS

Underprovision

Eric Gilbert pointed out the problem of underprovision in his paper "Widespread Underprovision on Reddit", which was evident by the fact that the pageviews of popular images are about two orders of magnitude of that of new images. We assumed the same problem would exist in Amazon movie reviews, so we compared the number of reviews each movie got, and sorted them. Here's the result:

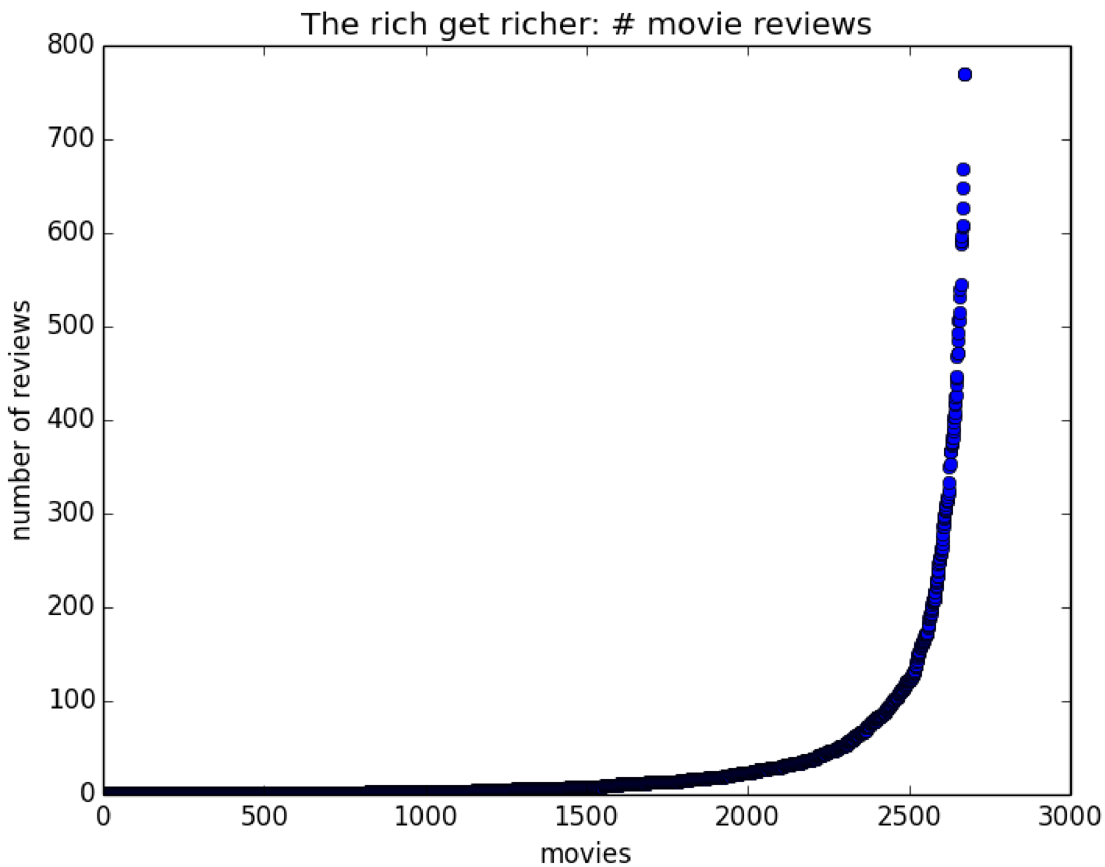

Figure 1. The numbers of reviews movies have, sorted by the numbers of reviews

Figure 1 strongly depicts a tall head and a long tail, which indicate inequality. The movie with the most reviews received 770 reviews, whereas 822 movies received less than 3 reviews. Do you want to guess which movie got the most reviews?

Ratatouille!

We then represent the inequality with a Lorenz curve. The Lorenz curve is used to represent inequality of the wealth distribution, here we use it to represent inequality of the popularity distribution. It shows for the bottom x% of movies, what percentage (y%) of the total number of reviews they have. The percentage of movies is plotted on the x-axis, the percentage of total number of reviews on the y-axis. A straight diagonal line shows perfect equality, the further the curve is to the line of perfect equality, the more inequality.

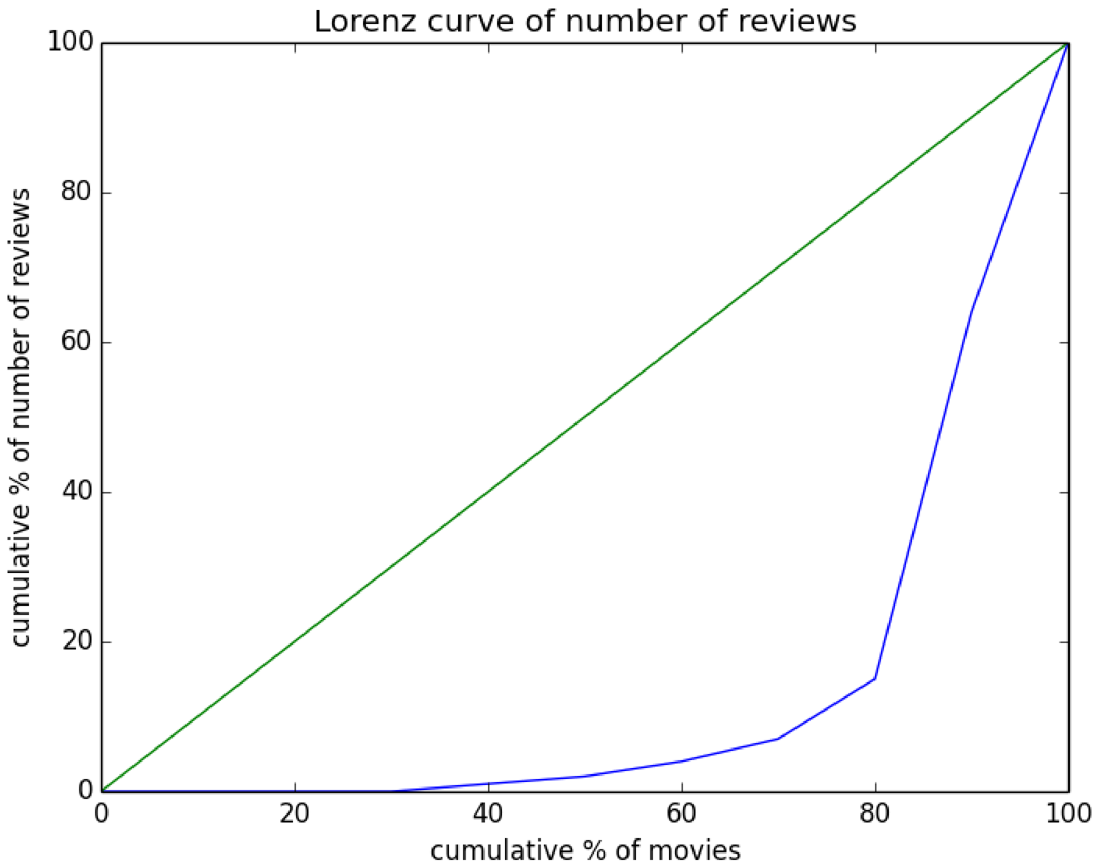

Figure 2. The Lorenz curve of number of reviews

We also looked at the inequality of the helpfulness distribution of reviews. For each review, users can vote whether it's useful or not. Some reviews that have a high number of engaged users who find it helpful, whereas others have none.

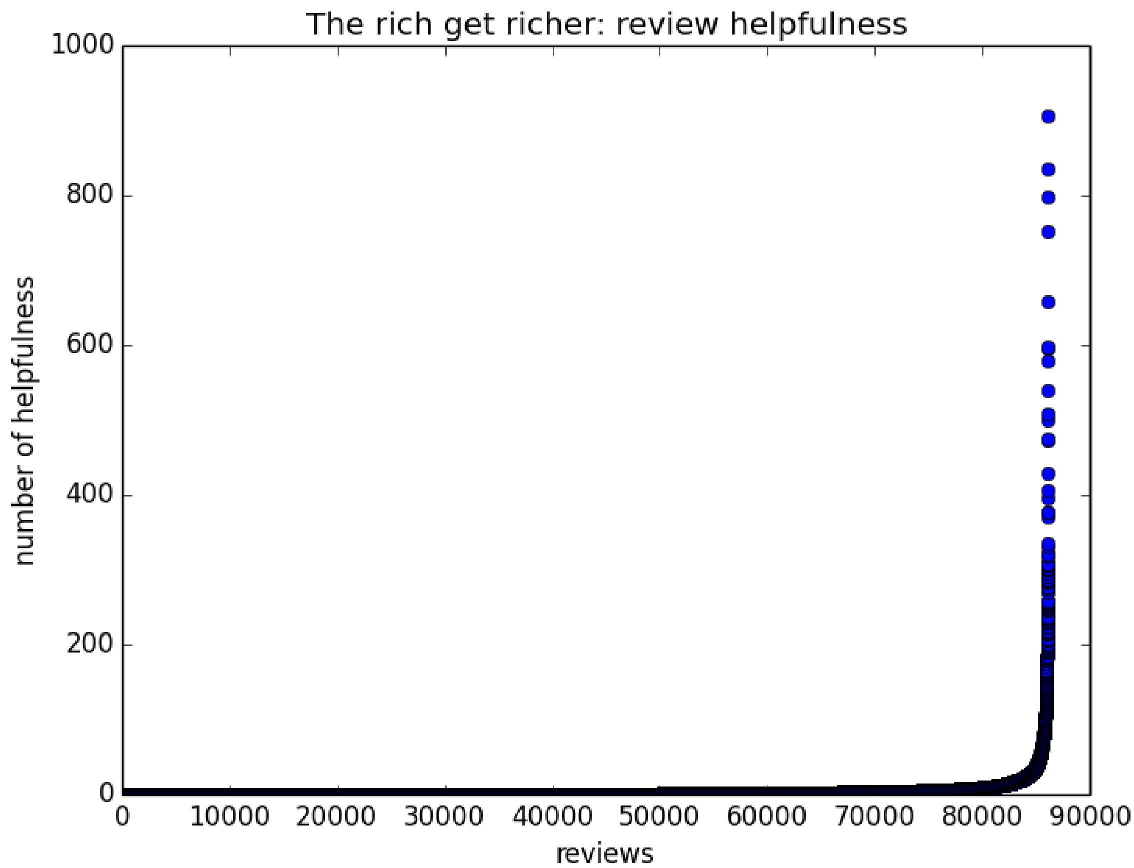

Figure 3. The helpfulnesses reviews have, sorted

The most helpful review has 957 users find it helpful, whereas there are 32446 reviews that no one voted on their helpfulness. The most helpful review was written by `jmastro`, for *Jillian Michaels 30 Day Shred.*

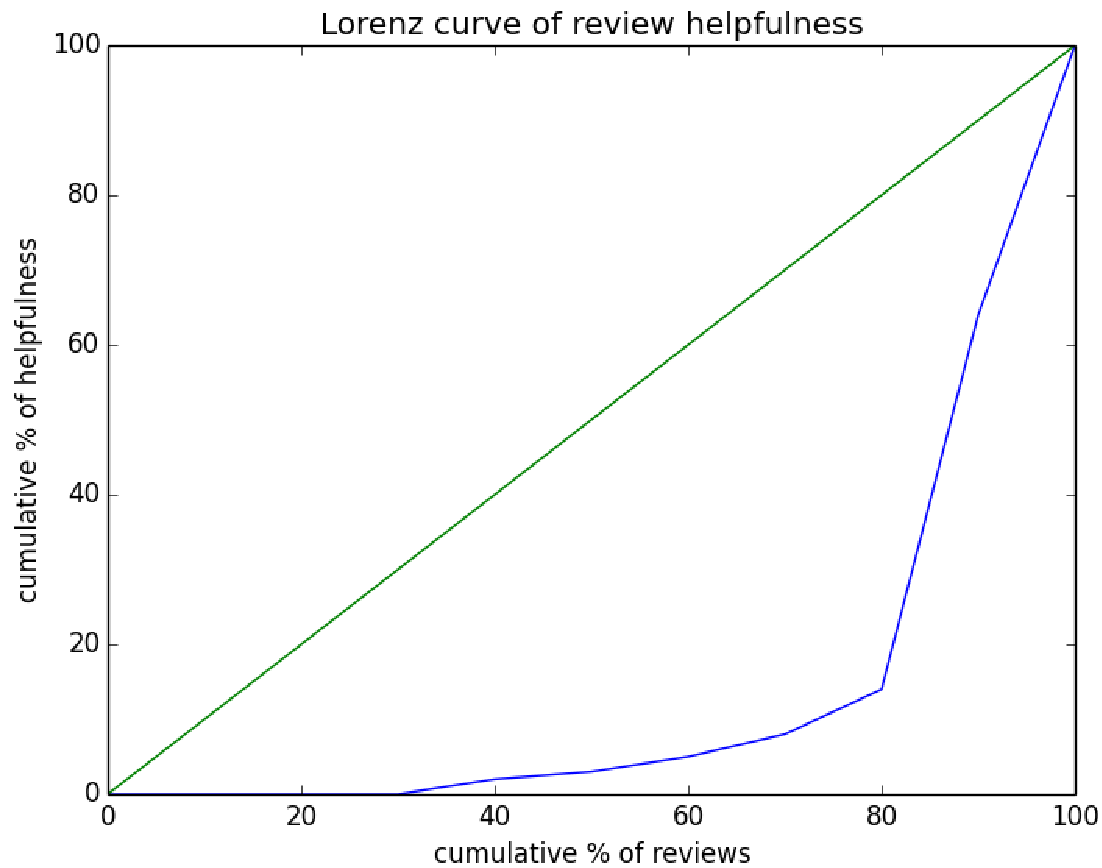

Figure 4. The Lorenz curve of review helpfulness

The Lorenz curve of review helpfulness shows considerable inequality over the distribution of help relative to the number of reviews. We contemplate that there are two reasons for this. One is that the inequality of review helpfulness is aggregated on top of the inequality of movie popularity, as a review of a popular movie (movie with greater number of reviews or page views) would get more views too, so more people would likely to vote on its helpfulness. Another reason is that three reviews of the greatest helpfulness are shown on the web page, whereas other reviews are hidden, so these three reviews would get great exposure than other reviews, hence they would get even more votes of helpfulness.

Perception

We want to prove that users have different ideas about the rating scale, for example, both user A and user B think that *Avengers* is a great movie, but user A would rate it 4 stars, while user B would rate it 5 stars. So we analysed the sentiment in reviews, as we assume a more positive review means more satisfaction with the movie. Then we tried to correlate the sentiment with the ratings, and see if there are any discrepancies. We used TextBlob API to analysis sentiment. The sentiment property returns a polarity score within the range of +1 to -1, +1 means absolute positive and -1 means absolute negative. The sentiment property also returns a subjectivity score, but we ignored it as movie reviews are subjective in general.

For each user, we got the polarity scores for each of his reviews, then averaged the scores; and we averaged the ratings he gave to each movie. Then we plot all users' average polarity scores on a scale of +1 to -1, and the average ratings on a scale of 1 to 5.

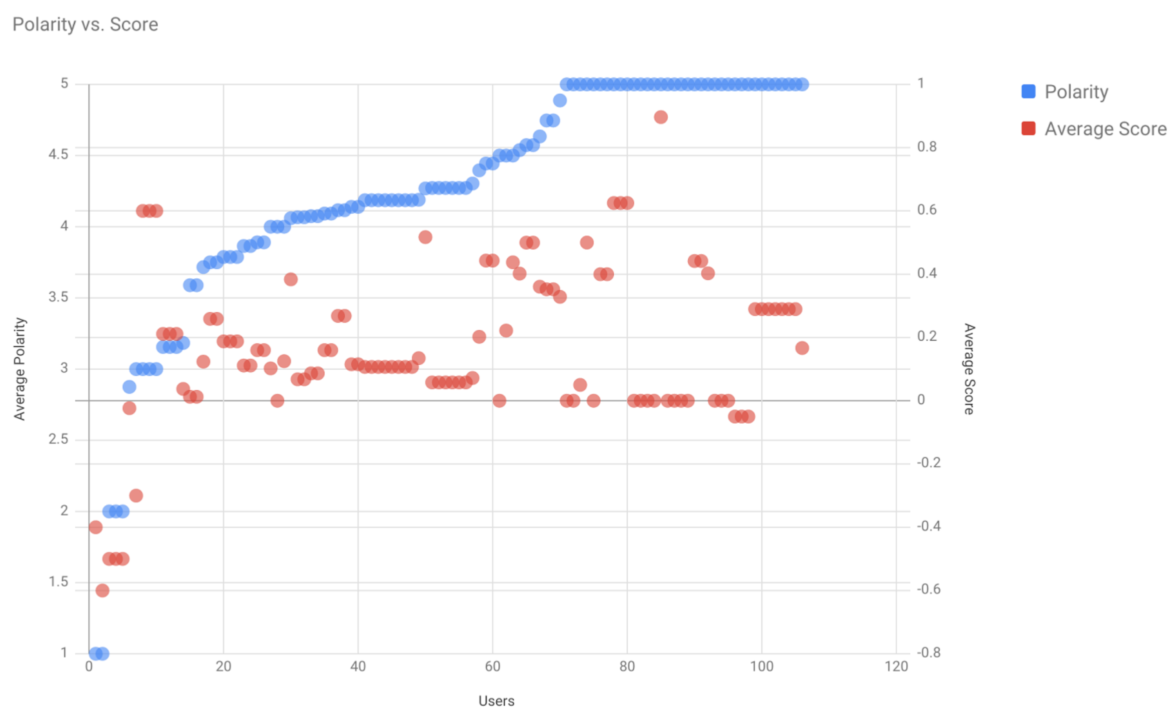

Figure 5. Review ratings vs. Review sentiment

The graph clearly shows no correlation between the polarity scores and the user ratings, which proves our point that users' sentiment about a movie has nothing to do with the ratings they give. However we do recognize the limitation of sentiment analysis, as it is not an accurate measure of emotion in text, it can't identify sarcasm, and users have different writing styles, the emotions in their review aren't necessarily proportional to their feelings about a movie.

Anchoring

By the theory of anchoring, previous reviews should have an effect on future reviewers answers. That is, if the average review score should have an influencing effect on the choice of a present reviewers score. We observed a range of adherence to this theory.

Looking at our results, we believe that there exists a threshold for movie reviews between strong subjective polarity and anchoring. If a movie to be controversial in depiction, its emotional viscerality may overpower effects from anchoring and result in no cohesive pattern of review ratings. We present here four examples within this range.

Example 1: Manhunter (1986)
Average rating: 3.8
reviews: 552

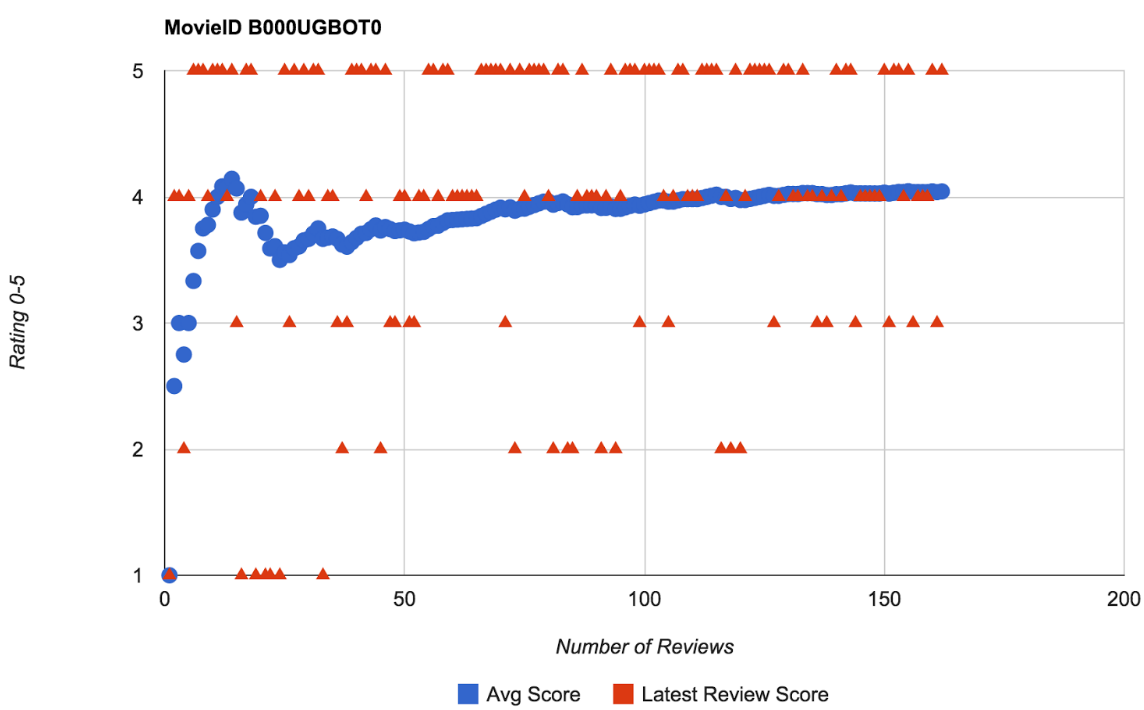

Figure 6. Strong Anchoring Example - Manhunter (1986)

We observe a convergence of future scores towards the mean value. Latest review scores form a concentration around the increasing mean with great clarity. The movie Manhunter 1986 is a critically acclaimed thriller with a very high score on rotten tomatoes of 94%. As an acclaimed thriller, we believe that many who watched it try to “fit in” with the experts: people that came before them; anchoring provides a strong draw as reviewers rate to feel relevant to a proclaimed masterpiece that they may not actually understand. We feel that this is the “Someone with great authority says its good, so it must be good,” example. Furthermore, as a moderately old movie, there has been time for the initial impressions and memories to wear off; the strongest remnants of its memory stand with its past ratings of critical acclaim.

Example 2: Red Dawn (1984)
Average rating: 4.1
\# reviews: 857

Figure 7. Medium Anchoring Example - Red Dawn 1984

Reviews of Red Dawn generally converge towards the established mean with less confidence than Manhunter. A lookup of Red Dawn finds that critics found it to be an average movie at 53% on Rotten Tomatoes. We believe that this means Red Dawn is a bit of an average movie that is moderately influenced by standing reviews before it and an individual taste. As a movie from a similar time as Manhunter, past public impressions have worn off. As critics did not proclaim a strong consensus either, we feel critic impressions have also worn off. Without past critical acclaim there is generally less tendency to adhere to a preconceived notion of what it should be, yet the movie itself is not polarizing enough in greatness or failure to overpower anchoring altogether.

Example 3: The Count of Monte Cristo (2002)
Average rating: 4.4
reviews: 1281

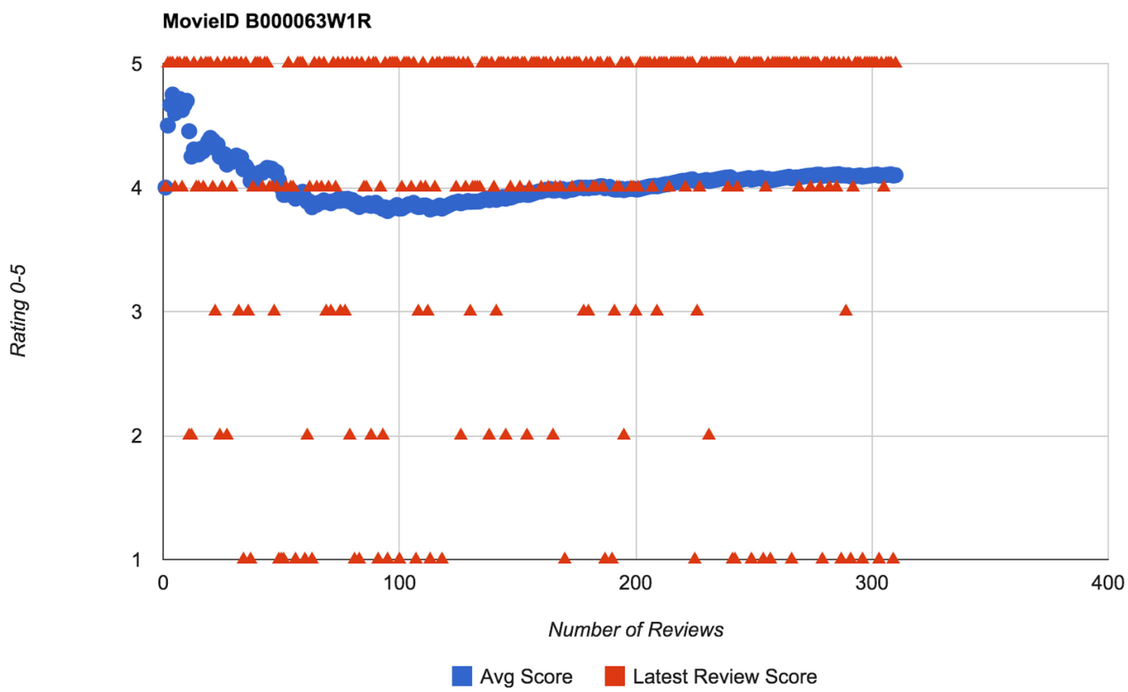

Figure 8. Medium Anchoring Example -The Count of Monte Cristo 2002

The Count of Monte Cristo provides an interesting case. It is a very popular movie amongst the public with a quaint simplicity. Rotten Tomatoes rates it as at 73%, a decent movie according to critics, and describes it as as an “old-fashioned yet enjoyable swashbuckler”. General popularity lends to a general convergence, but as a movie governed greater by personal experience and a simpler understandability rankings tend to maintain a degree of spread throughout. We observe a bimodal distribution of ratings with time; either reviewers side with the established anchored opinion or completely reject it as false. We feel that figure 8 suggests that the emotional pull of the movie is strong enough to break the threshold of anchoring with relative frequency.

Example 4: Terminator Salvation (2009)
Average rating: 3.7
reviews: 855

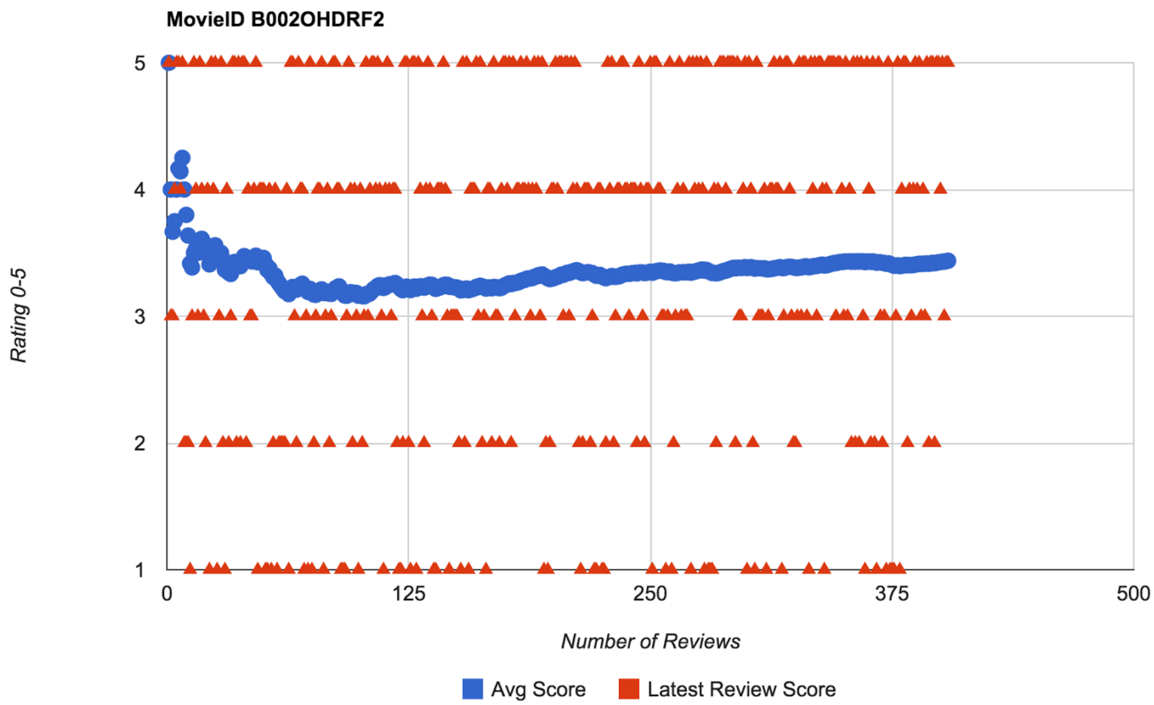

Figure 9. Weak Anchoring Example - Terminator Salvation 2009

Terminator Salvation shows very mixed opinions and no strong evidence of anchoring. The movie received a 33% score on Rotten Tomatoes. The movie itself includes a variety of draws in action, choice of actors, marketing, and context. Without an expectation to watch the movie, we believe that movie-watchers who tend towards anchoring, those who try to follow the experts would be less likely to watch this movie. The remaining audience of reviewers followed their impression experiences greater than previous trend; with no expectations the threshold falls in strong favor of emotional viscerality.

LAST NOTES

Our project is built on django, python, and MongoDB. We used python scripts to load and setup the database, mongodb to store and work with our approx. 7 million entries, and django to serve as a databridge to display the visualizations. Django and Mongo were linked via pymongo and mongoengine. The visualizations are refined using ZURB Foundation and charts were plotted using Google Charts API.

We had originally set up the project this way in anticipation of working with our very large dataset. In hindsight, a python notebook would have been much simpler. We made this project much harder than it should have been, and consequently have overshot the deadline. There was much time spent fighting the database and the limited restrictions of reasonable javascript chart performance for high numbers of points. We put a lot of work into this machine, really tried our best, and hope that our work comes through.

The original intent was to communicate this project through html; to get a better feel for our work we include a github link. The github readme describes important files and functions. The website for viewing visualizations is live at http://104.236.95.164:8001/.

GITHUB LINK

https://github.com/DVLevine/Plato-Movie-Matrix

WEBSITE LINK

http://104.236.95.164:8001/