# CSC111 Project 2 Proposal: Identifying Quality Films Through Amazon Review Metrics

Akram Klai, Reena Obmina, Edison Yao, Derek Lam

March 6, 2024

## Problem Description and Research Question

With countless new movies hitting the screens every year, it's like standing at an all-you-can-eat buffet without knowing where to start. How do you pick what's worth your time and money in a world overflowing with choices?

Enter online movie reviews. In the digital age, online movie reviews play a crucial role in guiding viewers' choices. Among these platforms is Amazon. Amazon isn't just a shopping giant; it's much more than that. Amazon carries a vast repository of movie reviews, presenting a unique opportunity for deep analysis.

Imagine trying to sift through endless streams of reviews, each with its own set of stars, helpful votes, pictures, and passionate dissertations on why a particular movie "changed [their] life" or how it was "two hours [they'll] never get back." Consequently, this overload of data may leave the average user lost, confused, and even frustrated.

Fear not, our project aims to develop a nuanced and multi-dimensional approach to analyze these reviews, leveraging a combination of factors to identify and recommend the best movies based on Amazon reviews. We're taking all those numbers, comments, pictures, and plotting them into an easy-to-understand graph. One can think of us as a personal movie concierge, guiding you to your next favorite film with just a few clicks.

There is no need for endless scrolling and movie night duds. Just you, a great movie, and perhaps a bowl of popcorn. Welcome to the future of movie selection, where every night can be the perfect movie night.

**Research Question: How can we analyze and integrate diverse review metrics from Amazon to identify and recommend the best movies?**

## Computational Plan

Our project incorporates a small subset of Amazon movie reviews, which includes various metrics such as the overall rating, number of helpful votes, verification status of the review, and the textual content of the reviews themselves. Each review is uniquely identified by an Amazon Standard Identification Number (ASIN). This dataset is sourced from `https://nijianmo.github.io/amazon/`.

Here's an example of a review from the JSON file (note that we are not utilizing all of the included information)...


```
{
    "overall": 5.0,
    "vote": "3"
    "verified": true,
    "reviewTime": "03 13, 2017",
    "reviewerID": "A1RAPP3YN10O53",
    "asin": "0005019281",
    "style": {"Format:": " DVD"},
    "reviewerName": "Amazon Customer",
```

```
    "reviewText": "We loved it."
    "summary": "Henry Winkler A Christmas Carol",
    "unixReviewTime": 1489363200
}
```


We will develop a graph in which each vertex symbolizes a movie. The edges connecting these movie vertices represent the relationships between them, such as shared genres or reviewers who have critiqued multiple films, indicating a connection. These connections will aid our users in discovering movies that match their interests. In addition to the criteria previously mentioned, the number of reviews a movie receives will be used to gauge its "popularity" within the graph. This means that movies with a higher volume of reviews will be positioned to reflect their increased visibility and interest among the audience. Concurrently, the accumulation of helpful votes and a high overall star rating will enhance a movie's standing in terms of "goodness" on the graph (please refer to the .jpg file attached on MarkUs). To calculate the "goodness" score of a movie, we will use a custom measure based on the rating called the "reputation factor" and multiply it by the number of helpful votes for each review to get the "goodness" score for each review, then take the average of the "goodness" scores based on the number of reviews it has. The "reputation factor" is represented as a number between -1 and 1 that is based on the rating (i.e. a rating of 5.0 and 1.0 will have a reputation factor of 1 and -1, respectively) This approach allows us to not only identify movies that are widely discussed but also to distinguish those that are highly regarded.

Our project entails a series of computational tasks. We will first implement algorithms for data filtering and transformation to manage the dataset's size, focusing on refining incomplete reviews and prioritizing the top 500 movies based on review volume. This process requires us to manually translate ASIN codes into movie titles to aid clarity and to add genres (we could not find a dataset that provides this information). In our efforts to recommend movies that align with users' preferences, we will calculate similarity scores (with metrics explained in the previous paragraph) and will include graph traversal techniques. Importantly, in our analysis, a helpfulness vote on a review won't be weighted as much as a unique review, ensuring that original content has a higher influence on our recommendations. Leveraging these algorithms will facilitate a more targeted examination of the connections between movies.

We plan to utilize the 'plotly' and 'networkx' libraries. A key feature of our visualization approach includes a quadrant-style graph that categorizes movies based on their popularity (x-axis) and review quality (y-axis), enhanced with functionality that emphasizes similar movies as users hover over a title. The choice of 'plotly' for visualization is driven by its superior capabilities in creating dynamic, interactive scatter plots, enabling a visually engaging exploration of the movies. Similarly, 'networkx' is selected for its strong support in graph manipulation, particularly in adding weights to arbitrary data that we plan to use, serving as an ideal framework for our project's underlying data structure.

# References


[1] "NetworkX Documentation." NetworkX. `https://networkx.org` (accessed March 6, 2024).

[2] "Plotly Python Graphing Library." Plotly. `https://plotly.com/python/` (accessed March 6, 2024).

[3] Jianmo Ni, "Amazon Review Data (2018)." (2018). Distributed by Jianmo Ni. `https://nijianmo.github.io/amazon/` (accessed March 6, 2024).