Movie Reviews: What's Worth Watching?

CIS 4930: Data Mining Proposal

Reece Boulware and Adam Sanchez

1. Research Question:

"What movies are worth watching?" This is the question we want to answer for this project. We are trying to determine what movies anyone would enjoy versus the movies they would most likely want to avoid or dislike, based on the best and worst movie that person's previously watched and their demographics. The goal is to allow users to look up movies they have never seen before and be shown how likely they are to enjoy it.

With how many movies are being made and released every day, we want users to be able to sift through the thousands of films they would have no interest in seeing and highlight the movies they will find worth watching based on other movies they enjoyed and rated well. We want to base these generated ratings on that user's previous ratings of films, including information about the film like any actors, directors, producers, and studio that might be shared with any other previous reviews.

2. Literature Survey:

The area of movie recommendation systems is a fairly established field. With the explosion of video streaming services in recent years, the goal of determining what a user might like based on their previous interest has become important. With this, and the expansive public databases provided by movie sites like IMDb and Rotten Tomatoes, there is quite a lot of data to work with online [7-9].

As stated in our introduction, our inspiration for this project lies in the issue we have of there being so many movies out there to pick from and potentially watch. Making a system that can tell us if a movie is not worth watching can save us time, and allow us to watch more movies we will enjoy. We also take inspiration from learning about current movie recommendation systems being used, both by researchers and big tech companies[4-6]. Although we will be implementing the same collaborative filtering techniques many of these systems use [1-6], our goal will be to give the user a rating (1-5) of what the system believes they will think of a particular movie. This differs from typical systems, who usually aim to return a list of movies the user might like [6].

3. Algorithm(s):

We plan to implement recommendation algorithms like matrix factorization and the Singular Value decomposition (SVD) algorithm. These are good for user-to-item recommendation algorithms, as the goal of matrix factorizations is to learn low-dimensional vectors for all users and items, where these vectors encode how much an item has a specific feature and how much a user enjoys those features, like how much a user likes or dislikes a

particular genre or director [1-3, 6]. The matrix comprises each row representing a user and each column the item. SVD is used as a collaborative filtering (CF) algorithm to predict the rating for a user-item pair based on the user and their top/bottom ratings, and the movie that they might enjoy [1-3, 6].

4. Expected experiments and analysis to be performed:

We plan to test if our film recommendations are accurate by gathering movie review data from the *GroupLens* 'MovieLens 1M' and 'MovieLens 100k' Datasets. The first dataset contains 1-million movie ratings of about 3,900 movies from just over 6,000 users, along with demographic info on each user(Gender, Age, Occupation, Zip-code) [8]. The second contains 100,000 ratings from 943 users on 1682 movies [7]. To populate each user's best and worst movies, we will pick 2 movies from each category based on their reviews. The users' best and worst movies will not be trained on those users. And with each user having a minimum of 20 reviews in each dataset, that brings our minimum number of reviews per user to 16. We will use this in addition to the IMDb movie database, which holds data from just about every movie made [9]. Together, data from both sets will combine to produce the dataset we will use for our project, with the target attribute being the users rating of the film.

We will then use part of our dataset to build our model, with the remaining used for testing purposes. After we get to a point where most of the previous test results come back accurate and we have a good model, we are planning to create a test using ourselves or our friends, where we take their demographics and best/worst films and apply them to some movies, which we will then watch or show that person and get what their actual review of that film was and compare that against the models prediction. We will implement this data into a matrix by having the user represent one vector and the item another; finding the dot product of the two leads to the expected rating of the items by the user.

5. Timeline:

First, we will collect and preprocess our data to get in how we would like. Then, we will structure the user and movie review data for matrix factorization and build the Python code. Then, we plan to test out smaller amounts of data first to ensure our programming is gathering the data we need. After we get the code functioning with a small amount of data, we will begin to implement the large sample of users and movies gathered from our dataset to test our recommendation system with much larger sample sizes and begin to predict movie receptions for these users, as well as test the reception against movies they had already left reviews for to see how accurate our system is. Afterward, we will try our code with someone who has done many reviews and compile their predicted reception for a movie they have never seen before. After finding that movie, we will have that person watch it and see how accurate our recommendation system was to the person's thoughts on the film.

References:

- [1] Contal, E. (2022, September 8). *What are the top recommendation engine algorithms used nowadays?*. Medium. https://itnext.io/what-are-the-top-recommendation-engine-algorithms-used-nowadays-646f588ce639
- [2] Chen, D. (2020, August 5). *Recommender System — Singular value decomposition (SVD) & truncated SVD*. Medium. Retrieved October 13, 2023, from https://medium.com/@m_n_malaeb/singular-value-decomposition-svd-in-recommender-systems-for-non-math-statistics-programming-4a622de653e9
- [3] Zhang, A., Li, M., Smola, A. J., & Lipton, Z. (2023). *Dive Into Deep Learning*. Cambridge University Press.
- [4] Vidiyala, Ramya. "How to Build a Movie Recommendation System?" Medium, Towards Data Science, 21 Oct. 2020, towardsdatascience.com/how-to-build-a-movie-recommendation-system-67e321339109.
- [5] Jayalakshmi S, Ganesh N, Čep R, Senthil Murugan J. Movie Recommender Systems: Concepts, Methods, Challenges, and Future Directions. Sensors (Basel). 2022 Jun 29;22(13):4904. doi: 10.3390/s22134904. PMID: 35808398; PMCID: PMC9269752.
- [6] Kniazieva, Yuliia. "Guide to Movie Recommendation Systems Using Machine Learning." High Quality Data Annotation for Machine Learning, Label Your Data, 14 Apr. 2022, labelyourdata.com/articles/movie-recommendation-with-machine-learning.
- [7] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19 (December 2015), 19 pages. DOI=http://dx.doi.org/10.1145/2827872
- [8] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19 (December 2015), 19 pages. DOI=http://dx.doi.org/10.1145/2827872
- [9] IMDb (n.d.). *IMDb Non-Commercial Datasets*. IMDb Developer. Retrieved October 13, 2023, from https://developer.imdb.com/non-commercial-datasets/