Movie Recommendation Model

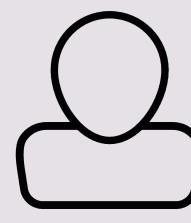
By Grant Doan, Prayash Joshi, Reagan Orth, Ved Patel

BACKGROUND

We aimed to create a recommendation system that considers user interests and similarities between users. Due to a lack of user input, existing approaches suffer from single-movie or search-based recommendations, popularity bias, and unreliable suggestions.







Limitations of the current solutions:

Cold Start: Not having any prior knowledge on a user



Data Sparsity: Difficult to find users who have reviewed the same movies



Scalability: Inaccurate results are inevitable with large amounts of data being processed on popular sites.



Filter Bubble: Users are diverged from having different opinions



We combat the issues with preexisting models by offering a way for users to enter a rating for movies directly. We will carefully evaluate their ratings through our machine-learning ecosystem to curate a set of movies they will surely enjoy.

OUR PROCESS

Initially, we started by scraping two different datasets from IMDb:

- Dataset 1: Reviews of top movies
- Dataset 2: Title, Description, Genre, Directors, and Actors of top 1000 movies

Now we had to perform NLP preprocessing on the first dataset which involved:
- Cleaning up punctuation and creating a sparse matrix on the reviews

- DF/TDF to remove insignificant words (Giving special attention to names mentioned in reviews)
- Pretrained BERT for Aspect Based Sentiment Analysis
- TF/IDF for finding specific film elements that are preferred by users

From the second data set we were able to create a similarity matrix which determines how close different movies are:

title	0	1	2	3	4	5	6	7	8	9
The Shawshank Redemption	34.0	28.0	8.0	12.0	50.0	0.81912434	0.8182012	0.81262195	0.81262195	0.81196874
The Godfather	40.0	5.0	198.0	26.0	89.0	0.86396104	0.8533033	0.8357353	0.8340453	0.8324910
The Dark Knight	125.0	59.0	38.0	154.0	36.0	0.9831174	0.9128396	0.87940204	0.87813926	0.8749045
The Lord of the Rings: The Return of the King	8.0	12.0	34.0	54.0	96.0	1.0354966	1.0354966	0.9367938	0.912353	0.8890849
Schindler's List	230.0	68.0	172.0	58.0	145.0	0.9251975	0.9199983	0.9156595	0.9155128	0.9130086
The Godfather Part II	1.0	85.0	55.0	79.0	14.0	0.8533033	0.84700835	0.8395854	0.8258126	0.8251418
12 Angry Men	45.0	82.0	154.0	79.0	178.0	0.80633354	0.8037954	0.8020277	0.80064553	0.79741997
Pulp Fiction	26.0	79.0	82.0	154.0	198.0	0.82960427	0.8246733	0.82354766	0.81916565	0.81760144
The Lord of the Rings: The Fellowship of the Ring	12.0	3.0	34.0	54.0	172.0	1.100281	1.0354966	0.94287175	0.9245937	0.9231339
Fight Club	21.0	219.0	172.0	43.0	4.0	0.87493265	0.8421202	0.83727264	0.8361622	0.8226889

Next we created a Graph Neural Network to create a high dimensional mapping for the movies:

- The graph itself is large and unweighted
- Movies are nodes
- Edges represent similarities
- Similar movies will be close together



But how will this help recommend movies?

- Check movies most similar to movies the user already likes
- Check descriptive words in any user reviews provided
- Combine BERT pretrained-model and TF/IDF to find what user's find interesting in movies

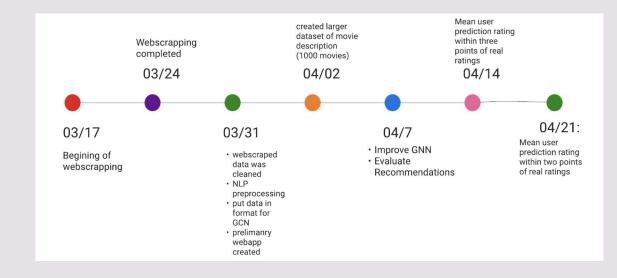
And finally combine the movie preferences...

Results and Impact

Graph Neural Network:

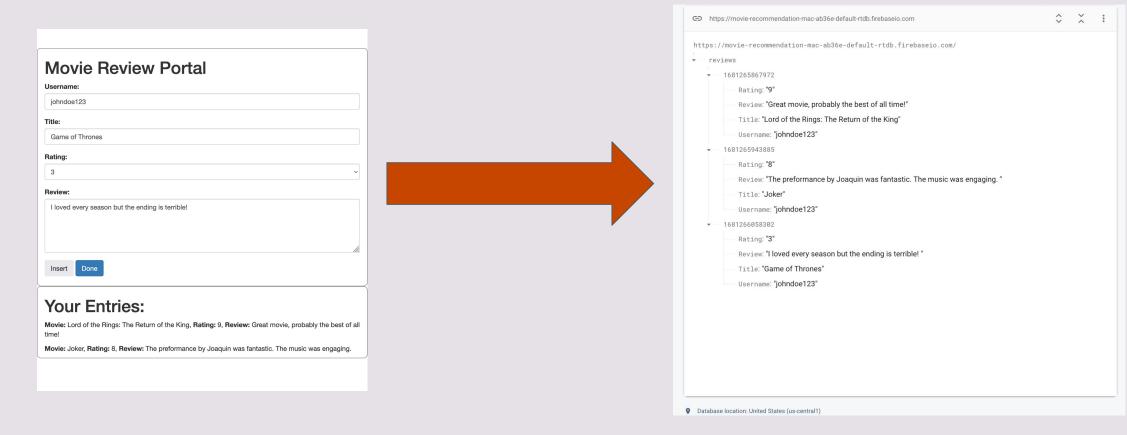
- Built similarities graph for unsupervised problem.
- Worked on improvements such as gathering more film data.
- Pretrained BERT model does a good job of evaluating overall sentiment with 90% accuracy.
- Extracting topics from reviews is difficult, but positive/negative classification works well.
- Pursued aspect-based sentiment analysis using PyABSA.

Here is our project timeline:



Webpage:

- Takes in user reviews and provide recommendations based on them.
- First iteration took one review at a time and stored in Firebase real-time database.
- Second iteration allows for input of multiple movies and transitioned to a dynamic website.



Suppose you entered this review for The Lord of the Rings: The Return of the King:

"Rating: 9"

"I love this movie. It has amazing world building and the characters are played very well. I wish more movies were like this."

This review would result in these results:

