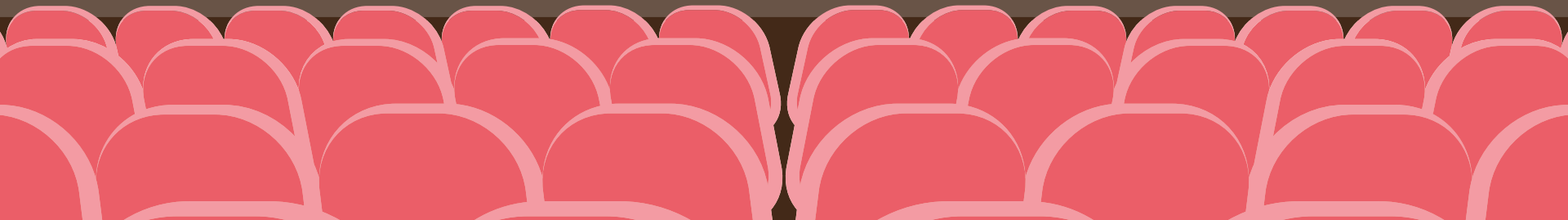
FILM RECOMMENDATION!

Everett Hayes, Rayaan Attari, Himanshu Bainwala

CSC-207 Final Project

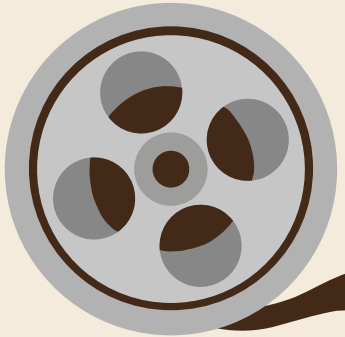


INTRODUCTION

- Due to required quarantines and social distancing, films have become an even more important part of our lives!
- With an ever increasing amount of streaming services and content available, how do we choose what we watch?
- Our project will streamline this process for users.
- Users can enter movies they like and our algorithm will handpick movies we think will be to their liking.

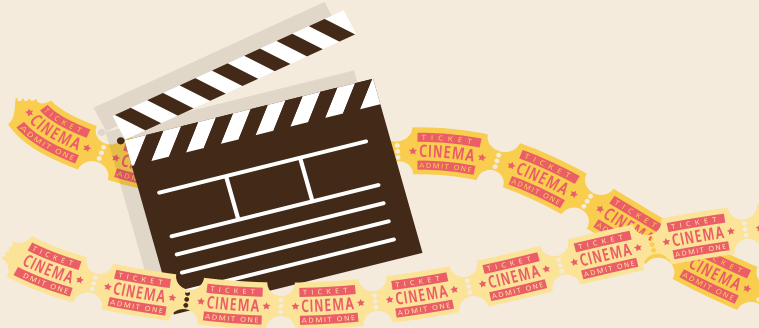
DATASET

- Our data comes from IMDB and “The Movie Database”
- Metadata on ~5000 films
- We converted from CSV → JSON
- Our algorithm uses: Keywords, Genres, Production companies, IMDB score, Audience Vote, and Popularity



DATA CLEANING AND SETUP

- First we parsed the JSON object for each film into a class we created using the JSON-simple library.
- The string fields contained nested JSON objects, so we extracted only the text fields as required.
- We also normalized all of the numerical fields, meaning we replaced them with z-scores for easy comparison among fields. (This doesn't affect underlying distribution)

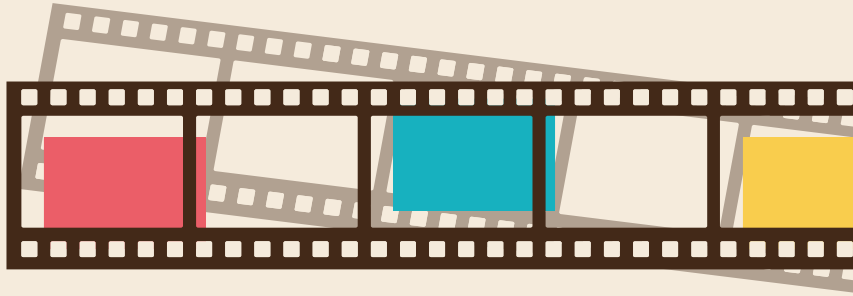


USER INPUT

- Each round the user is offered a list of films. They can either choose one they like or pass.
- This continues until the user has chosen 3 films they like.
- We then approximate what the user would like based on their selected films.

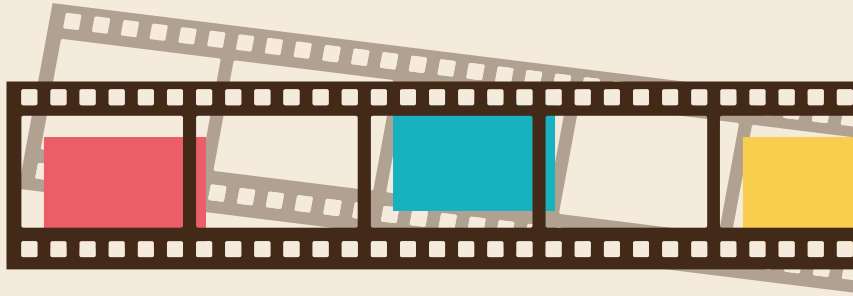
RECOMMENDATION ALGORITHM - NUMERICS

- This approximation of the user's choices is then compared to every movie in the data.
- For the numeric fields, this is relatively simply as we are simply trying to minimize the difference between a film and this approximation



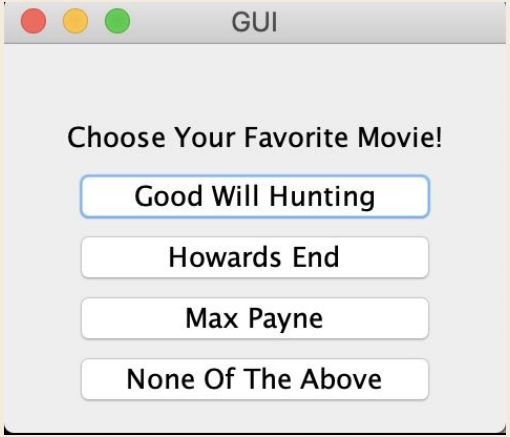
RECOMMENDATION ALGORITHM - STRINGS

- We then implemented the Jaro-Winkler distance algorithm, used to compare string similarity.
- We compare the approximation of the user's choices, with the fields of every film. Meaning ever approximation keyword is compared with every movie keyword, and so on for genres, and production company.



OUR PROGRAM IN ACTION!

- Each round looks like this
- After they pick 3 movies
- We then present the user with the most similar movies



THE FUTURE

- At first our keyword analysis was much more computationally intense. We relied on cosine similarity using vector analysis to find the most interrelated movies. The computation time was long (like really long), however we found the results were more accurate.
- We excluded this part of the algorithm from the final project, (the code is still included however) however in the future if we can get the computation time down it will be a valuable part of our project.