

Movie Review System - Documentation

Team Charaka

ANIRUDH SRINIVASAN

CS20BTECH11059

CHIRAG MEHTA

AI20BTECH11006

RUTV KOCHETA

MA21BTECH11014

PRANAV BALASUBRAMANIAN

AI21BTECH11023

SURYAANSR JAIN

CS21BTECH11057

DISHANK JAIN

AI20BTECH11011



Contents

1	Our Idea	3
1.1	USP (Unique Selling Point)	3
1.2	Motivation	3
2	Development	4
2.1	Frontend	4
2.1.1	Landing Page	4
2.1.2	Movie page	4
2.2	Backend	4
2.2.1	APIs	4
3	ML	5
3.1	Data	5
3.1.1	Context	5
3.1.2	Content	5
3.1.3	Acknowledgement	5
3.1.4	Inspiration	5
3.1.5	Attribute Details	5
3.1.6	EDA	7
3.2	Summarization	15
3.3	Sentiment Analysis	16
4	References	16

1 Our Idea

The project revolves around the development of a Movie Rating and Review System. This system offers a comprehensive platform for movie enthusiasts to create accounts, access information about movies, share their opinions through reviews and ratings, and engage in various movie-related activities. It incorporates cutting-edge technologies such as NextAuth for user authentication, autocomplete search for movie discovery, machine learning-generated text based opinion summarization for reviews, and sentiment analysis-based clustering of audience reviews.

1.1 USP (Unique Selling Point)

The uniqueness of this Movie Rating and Review System lies in its integration of advanced technologies to enhance the user experience. Key features that set it apart from traditional movie review platforms include:

1. **Review and Rating Capabilities:** Users can share their thoughts on movies, giving ratings and reviews, with the flexibility to edit or delete their contributions.
2. **ML-Generated Opinion Summarization:** The system employs machine learning to automatically summarize user reviews, providing concise and informative insights into a movie's reception.
3. **Sentiment Analysis-Based Clustering:** Audience reviews are clustered based on sentiment analysis, categorizing them into positive, negative, and neutral, facilitating quick comprehension of public opinion.
4. **NextAuth Integration:** Users can easily login from accounts through NextAuth, ensuring a secure and hassle-free experience.
5. **Autocomplete Search:** The system offers a user-friendly movie search feature with autocomplete suggestions, streamlining the process of discovering films.

1.2 Motivation

The motivation behind developing this Movie Rating and Review System stems from several factors:

1. **User Empowerment:** Enabling users to express their opinions and read others' reviews empowers them in making informed decisions about which movies to watch.
2. **Data-Driven Insights:** By utilizing machine learning for text summarization and sentiment analysis, the system offers data-driven insights into movies' reception, aiding both users and filmmakers.

3. **Ease of Use:** Incorporating NextAuth and autocomplete features simplifies the user experience, making it more accessible to a wider audience.
4. **Community Building:** The platform fosters a sense of community among movie enthusiasts by providing a space for discussion and interaction.
5. **Continuous Improvement:** Regular updates and improvements ensure that the system remains current and aligned with user expectations in the dynamic world of cinema.

2 Development

2.1 Frontend

2.1.1 Landing Page

Landing page has the grid of all movies fetched from backend with pagination implemented and user can also search for movies using a autocomplete item in the header. The header also contains the profile image. User can see if they are logged in or not on clicking the profile image. Clicking on any movie tile or grid item the user will go to another page containing individual movie details.

2.1.2 Movie page

This page contains the name, the overview, the image, all the reviews of the movie. Users can add a review to the movie and the review goes for processing to the backend and the users can also see the sentiment analysis on the frontend. users can also filter the reviews based on the sentiments(positive, neutral, negative). Adding of the comment only happens if the user is logged in.

2.2 Backend

2.2.1 APIs

1. **GET /names:** gets all movies names and IDs (for search autocomplete)
2. **GET /search?q=:** search by title
3. **GET /movies?sort=&page=:** Sorting and pagination
4. **GET /movies/id:** Gets movie details by ID
5. **GET /movies/id/reviews:** gets reviews for movie
6. **GET /movies/id/reviews:** posts review of user for a movie (with auth)
7. **GET /movies/id/reviews:** delete review of user for a movie (with auth)

3 ML

3.1 Data

3.1.1 Context

Movies' data is stored on several popular websites, but when it comes to critic reviews there is no better place than Rotten Tomatoes. This website allows to compare the ratings given by regular users (audience score) and the ratings given/reviews provided by critics (tomatometer) who are certified members of various writing guilds or film critic-associations.

3.1.2 Content

1. In the movies dataset each record represents a movie available on Rotten Tomatoes, with the URL used for the scraping, movie tile, description, genres, duration, director, actors, users' ratings, and critics' ratings.
2. In the critics dataset each record represents a critic review published on Rotten Tomatoes, with the URL used for the scraping, critic name, review publication, date, score, and content.

3.1.3 Acknowledgement

1. Data has been scraped from the publicly available website [Rotten Tomatoes](#).
2. Since the data takes a few days to be scraped from the website, there is no full consistency between some fields of the movies and critics datasets such as "tomatometer_count", "tomatometer_top_critics_count", "tomatometer_fresh_critics_count", and "tomatometer_rotten_critics_count" of the movies dataset compared to all records included in the critics dataset, which has been scraped first.

3.1.4 Inspiration

To provide detailed information in regards to movies' critic reviews and the users' vs critics' ratings, and it can be combined with other movie datasets publicly available (FilmTV, etc.).

3.1.5 Attribute Details

1. rotten_tomatoes_critic_reviews.csv
 - (a) rotten_tomatoes.link: Link from which the movies data have been scraped - e.g. the record "m/0814255" has been scraped from "<https://www.rottentomatoes.com/m/0814255>"
 - (b) critic_name: Name of critic who rated the movie

- (c) `top_critic`: Boolean value that clarifies whether the critic is a top critic or not
- (d) `publisher_name`: Name of the publisher for which the critic works
- (e) `review_type`: Was the review fresh or rotten?
- (f) `review_score`: Review score provided by the critic
- (g) `review_date`: Date of the review
- (h) `review_content`: Content of the review

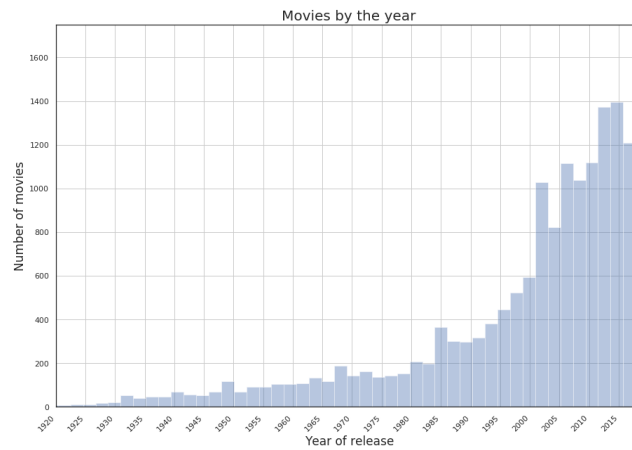
2. `rotten_tomatoes_movies.csv`

- (a) `rotten_tomatoes_link`: Link from which the movies data have been scraped - e.g. the record "m/0814255" has been scraped from "<https://www.rottentomatoes.com/m/0814255>"
- (b) `movie_title`: Title of the movie as displayed on the Rotten Tomatoes website
- (c) `movie_info`: Brief description of the movie
- (d) `critics_consensus`: Comment from Rotten Tomatoes
- (e) `content_rating`: Category based on the movie suitability for audience
- (f) `genres`: Movie genres separated by commas, if multiple
- (g) `directors`: Name of director(s)
- (h) `authors`: Name of author(s)
- (i) `actors`: Name of actors
- (j) `original_release_date`: Date in which the movie has been released
- (k) `streaming_release_date`: Date in which the movie has been released for streaming
- (l) `runtime`: Movie runtime (in minutes)
- (m) `production_company`: Name of the production company
- (n) `tomatometer_status`: Tomatometer value of "Rotten" (less than 60% positive reviews), "Fresh" (at least 60% of positive reviews), and "Certified Fresh" (at least 75% of positive reviews, at least 80 reviews of which at least 5 from top critics)
- (o) `tomatometer_rating`: Percentage of positive critic ratings
- (p) `tomatometer_count`: Critic ratings counted for the calculation of the tomatometer status
- (q) `audience_status`: Audience value of "Spilled" (less than 60% of users gave a rating of at least 3.5) or "Upright" (at least 60% of users gave a rating of at least 3.5)
- (r) `audience_rating`: Percentage of positive user ratings
- (s) `audience_count`: User ratings counted for the calculation of the audience status

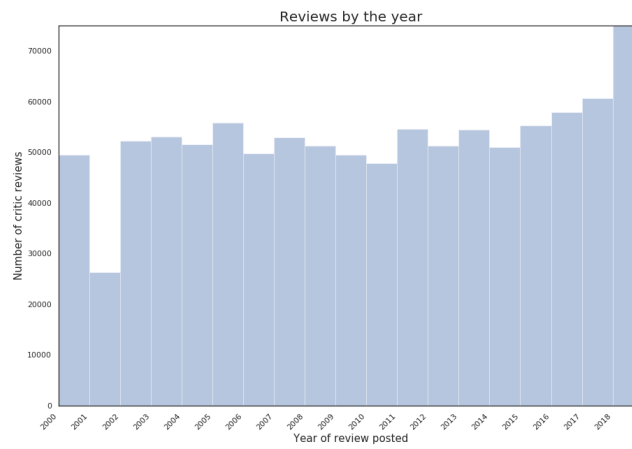
- (t) tomatometer_top_critics_count: Count of top critic ratings
- (u) tomatometer_fresh_critics_count: Count of fresh critic ratings
- (v) tomatometer_rotten_critics_count: Count of rotten critic ratings

3.1.6 EDA

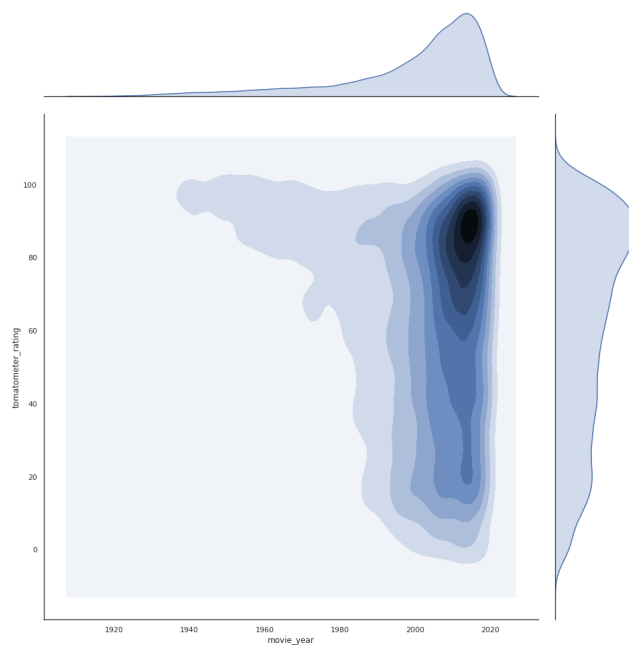
1. Histogram of Movies by year of release



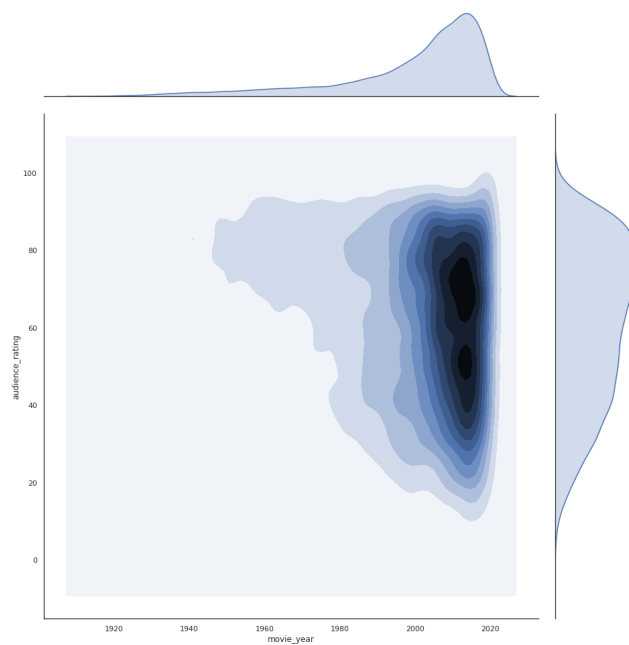
2. Histogram of Reviews by year of posting



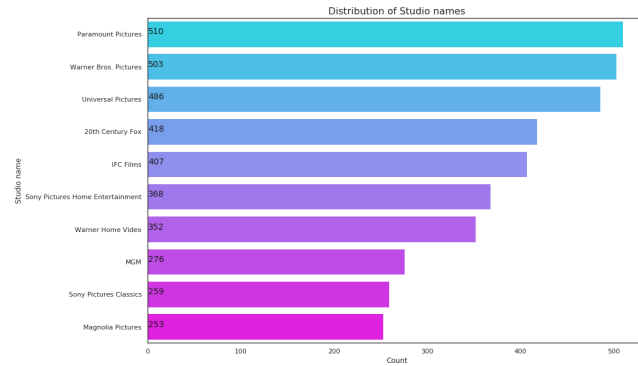
3. Distribution of TomatoMeter ratings across the years



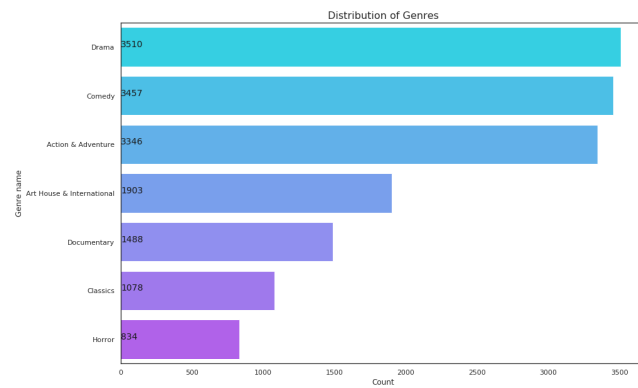
4. Distribution of Audience ratings across the years



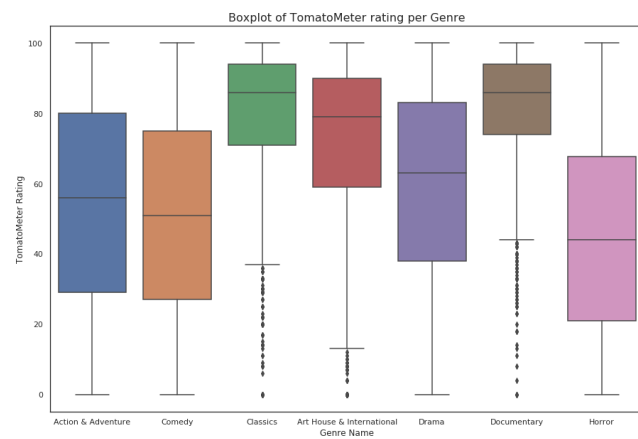
5. Frequency of Studio names

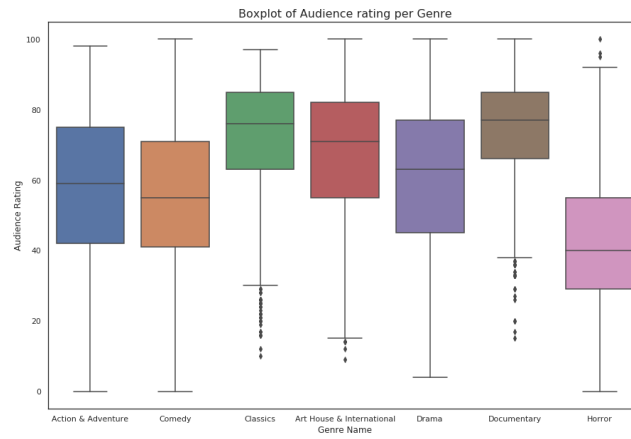


6. Frequency of movie genres



7. Boxplot of TomatoMeter and Audience ratings grouped by genre



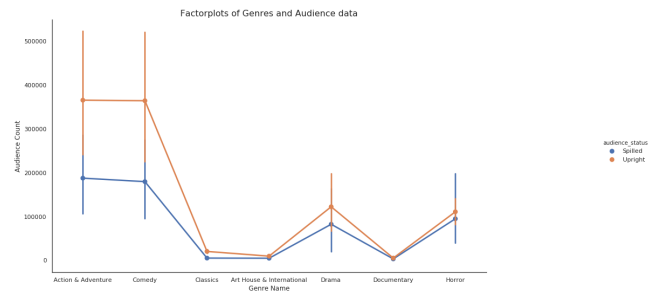
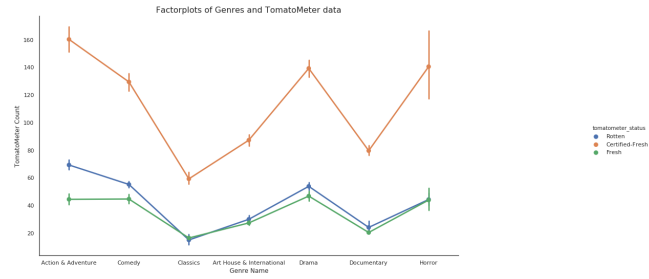


8. Crosstab of genres and TomatoMeter and Audience status

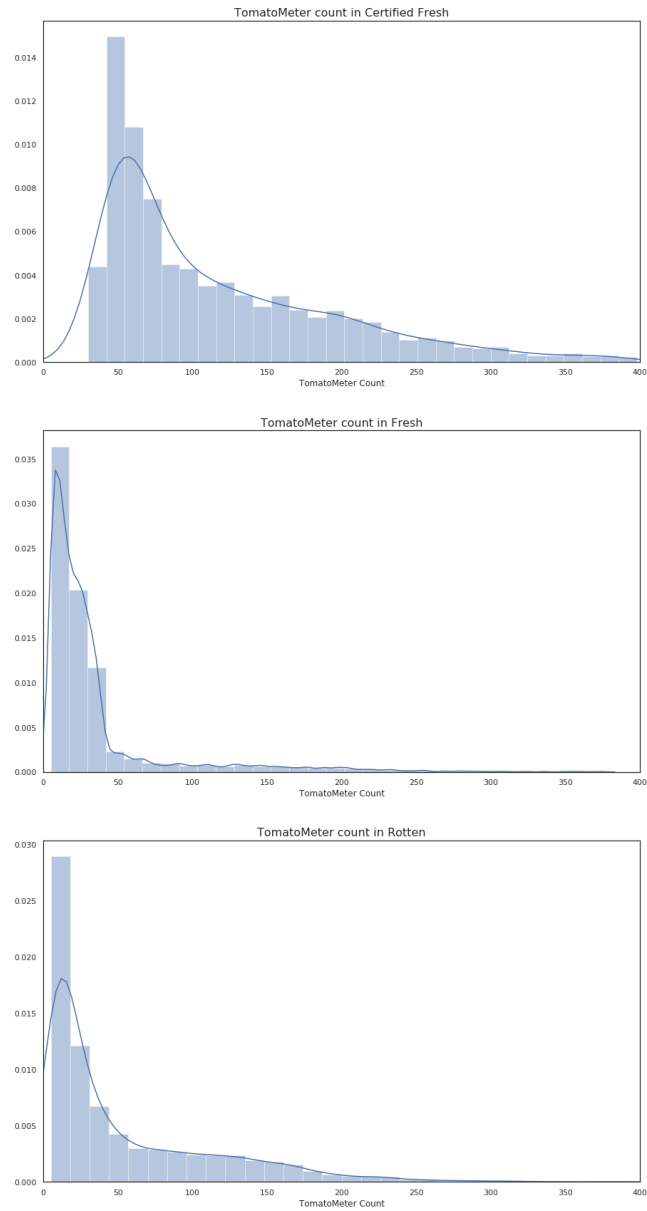
tomatometer_status	Certified-Fresh	Fresh	Rotten	All
first_genre				
Action & Adventure	541	1034	1771	3346
Art House & International	474	947	482	1903
Classics	153	764	161	1078
Comedy	559	892	2006	3457
Documentary	401	920	167	1488
Drama	824	1094	1592	3510
Horror	76	206	552	834
All	3028	5857	6731	15616

audience_status	Spilled	Upright	All
first_genre			
Action & Adventure	1700	1596	3296
Art House & International	581	1320	1901
Classics	229	849	1078
Comedy	1965	1469	3434
Documentary	247	1233	1480
Drama	1554	1912	3466
Horror	669	148	817
All	6945	8527	15472

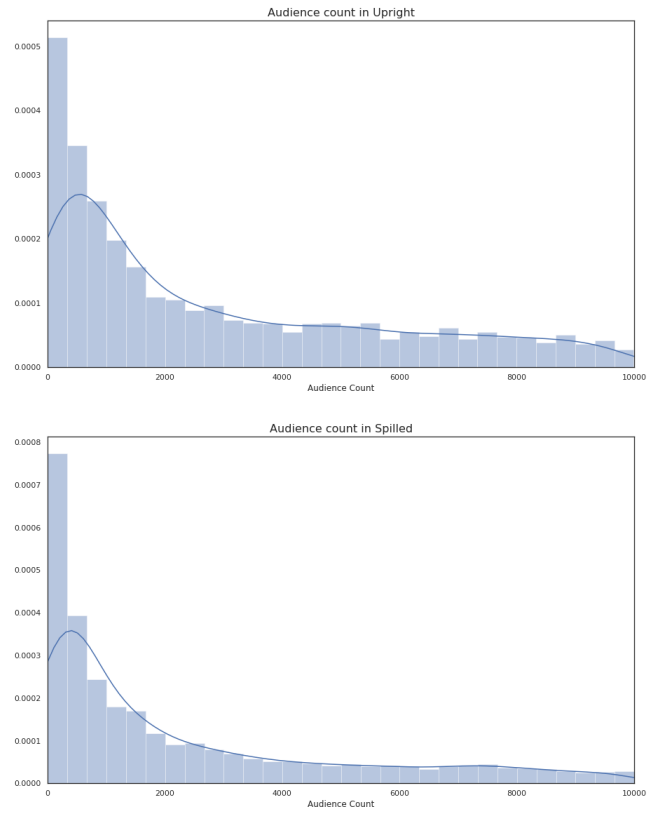
9. Factorplots of Genres and TomatoMeter and Audience status



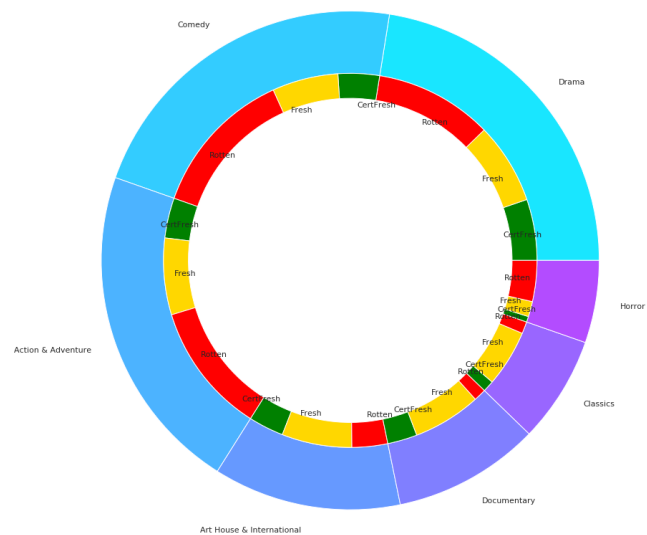
10. Distribution of TomatoMeter count per TomatoMeter status



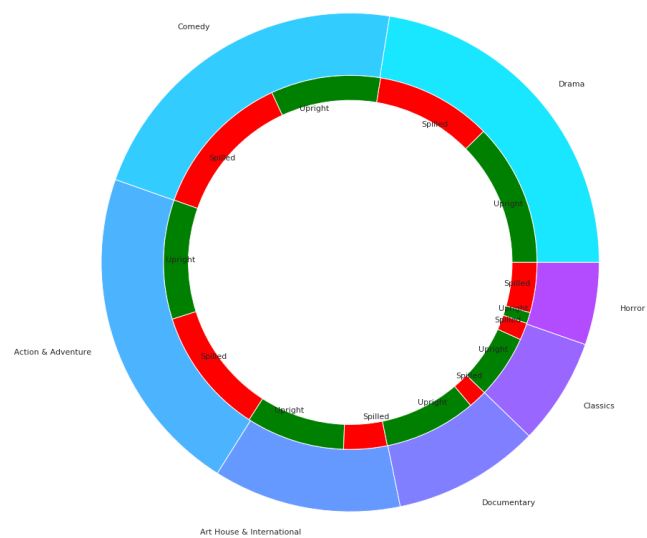
11. Distribution of Audience count per Audience status



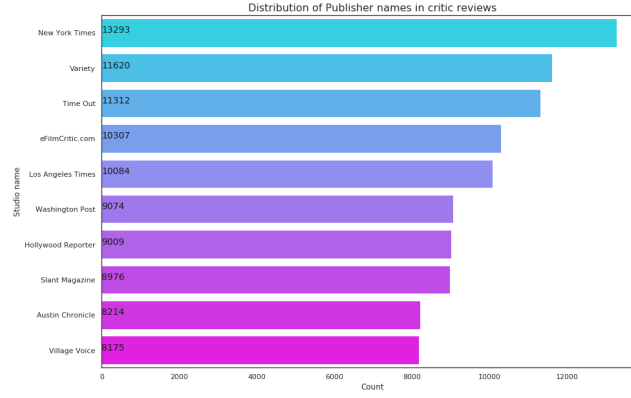
12. Pie chart of Genres and relative TomatoMeter status



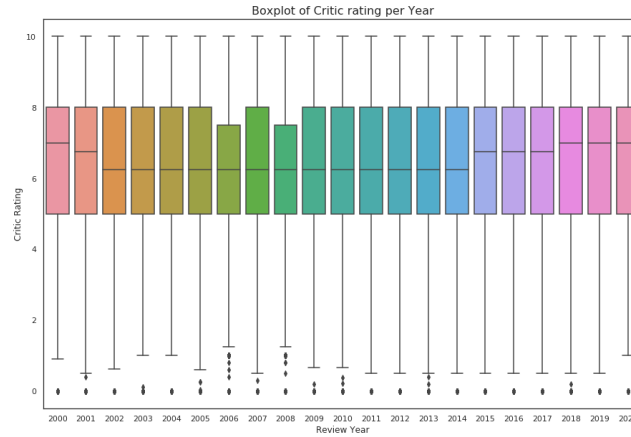
13. Pie chart of genres and relative Audience status



14. Frequency of Publisher names in critic reviews



15. Boxplot of Critic rating by year



3.2 Summarization

We use the "Unsupervised Opinion Summarization with Content Planning" paper to create a tool that summarizes a list of reviews for a given movie. It aims to provide concise and informative summaries of movie reviews, making it easier for users to get a quick understanding of public opinion about a film. The architecture of the model is given below:

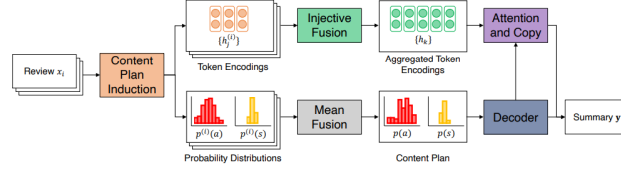


Figure 1: Model architecture of PLANSUM. The content plan is constructed as the average of the aspect and sentiment probability distributions induced by the content plan induction model. It is then passed to the decoder, along with the aggregated token encodings to generate the summary.

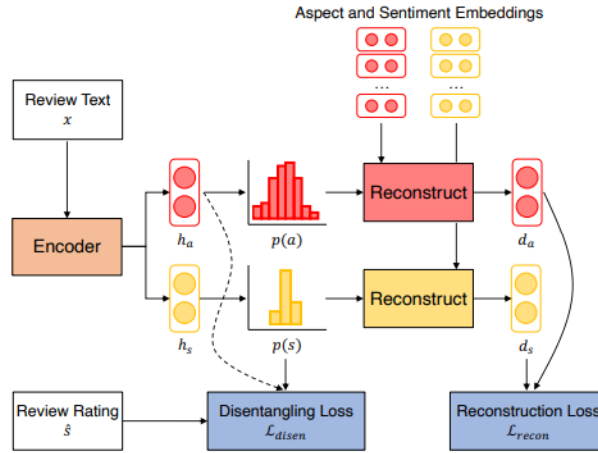


Figure 2: Model architecture of our content plan induction model. The dotted line indicates that a reverse gradient function is applied.

3.3 Sentiment Analysis

We use Roberta model to catogorize reviews as

1. Positive
2. Negative
3. Neutral

The model that we used is *cardiffnlp/twitter-roberta-base-sentiment-latest*

4 References

1. [Rotten Tomatoes - Movies and Critic Reviews Dataset](#)

2. [Unsupervised Opinion Summarization with Content Planning](#)
3. [TimeLMs: Diachronic Language Models from Twitter](#)
4. [TWEETEVAL: Unified Benchmark and Comparative Evaluation for Tweet Classification](#)
5. [SemEval-2017 Task 4: Sentiment Analysis in Twitter](#)