Is it *cinema*?

DATA 515: Software Design for Data Scientists Final Project

Dyuti Vartak
Edgar Lopez
Riyosha Sharma
Saikiran Babu

Background

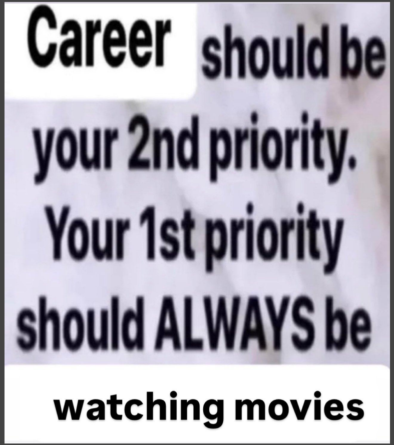
Problem: Do you really want to spend $15 dollars watching the new release in a theater or would you rather just watch it at 2x speed from your sofa?

Background

Solution: We can help! We built a webtool that -

- Given a Letterboxd movie link and username, provides:
  - a summary of the most popular reviews
  - an aspect based sentiment analysis of the most popular reviews
  - a personalized taste-match by comparing your reviews with the film reviews

- Given a Letterboxd username, provides:
  - a roast analyzing your film preferences



Use Cases

- **#1 Film Club Screening**

- **Want** to pick a horror-comedy for Halloween screening
- **Need** to understand
  - reviewers’ take
  - top positive/negative aspects of the movie
- No **technical skills** required

Use Cases

- **#2 Film Producer**

- **Want** a sentiment based analysis of the most liked/disliked aspects of their film
- **Need** to understand
  - top positive/negative aspects of the movie
- No **technical skills** required

Use Cases

- **#3 Movie Night with the Gang**

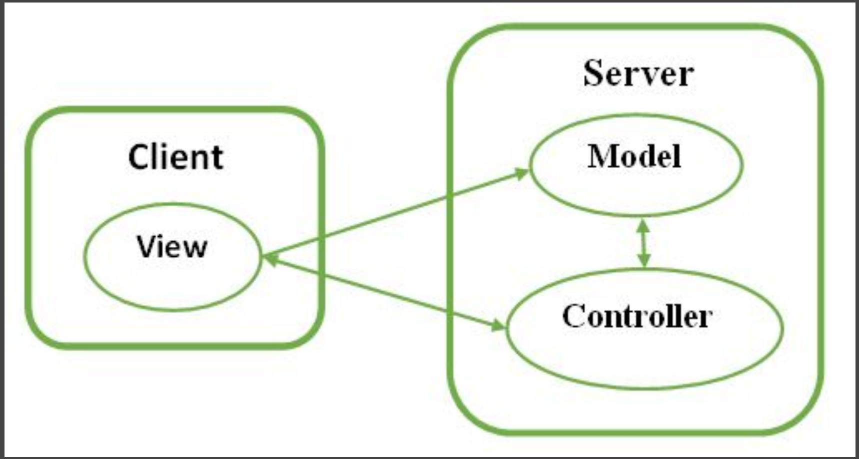
- **Want** to pick a movie EVERYONE likes
- **Need** to understand
  - top positive/negative aspects of many movies
  - film preferences of each friend
- No **technical skills** required

Data Used

- Scraped dynamically from letterboxd film/user link
    - Film link: 300 most popular reviews scraped
    - User profile link:
        - Review data scraped: movie name, rating, watched date, review
        - Stats data scraped: # films, # hours, # directors, # countries, longest watch streak and more

Project Architecture

MVC + Client-Server



Model: **Scrapers**+Analyzers

- Movie:
  - Name
  - Genres
  - Image
  - Release Year
  - ~ 360 most popular reviews
- User Profile:
  - Reviews
  - No. of movies watched
  - No. of countries
  - Total hours watched
  - Longest watch streak

Components: Scrapers+Analyzers

- Classes of generative functions to generate summaries, aspect charts, and the roast

- **Input:** Python data structures (scraped movie and user data)
- **Output:**
  - Movie reviews’ summary
  - Reviews’ aspect data
  - User roast
  - Vibe match summary

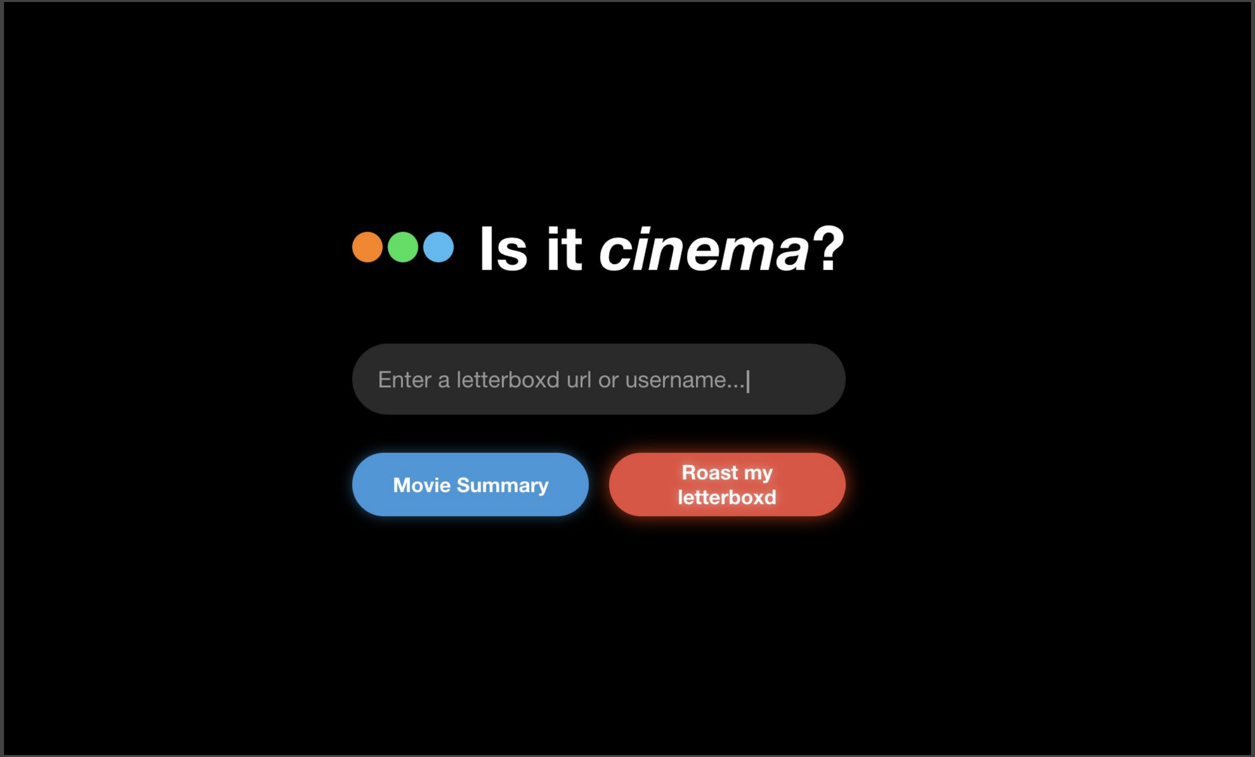
View: User Interface

- Web Application built using JavaScript and React
  - Node.js, Vite dev framework, React reusable components libraries
  - Asynchronous data fetching
  - Lego block building

- **Input:** Letterboxd Movie url and/or Username

- **Output:** movie details
  - Reviews’ summary
  - Sentiment-based aspect analysis charts
  - Vibe match analysis
  - User Profile Roast

View: User Interface



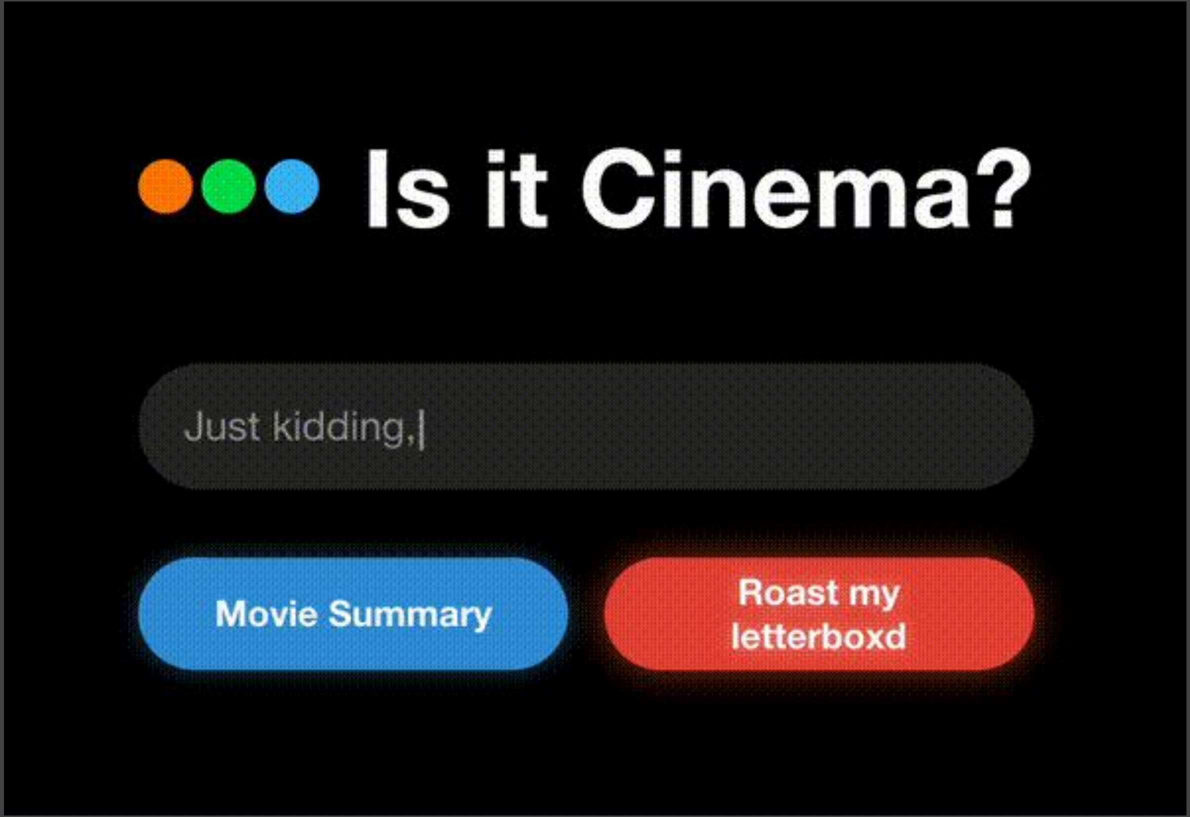
Controller

- Flask API for connection between source URL and Views

- 3 endpoints:
    - movie_details - returns details about the movie
    - roast - roasts the user based on their Letterboxd history
    - taste - tells whether a particular movie is of the user’s taste

- Each endpoint returns output in JSON format

Demo!!



Is it Cinema?

Just kidding,|

Lessons Learned and Future Work

- Deployment!
- Use LangChain and RAG models to make site conversational
- Git config on day 1 and not T-1 (;-;)
- Test-driven development
- Our application is dependent on Letterboxd site formatting
- Docker!