CSC111 Project 2: Identifying Quality Films Through Rotten Tomatoes' Metrics

Akram Klai, Reena Obmina, Edison Yao, Derek Lam

April 1, 2024

Introduction

With countless new movies hitting the screens every year, it's like standing at an all-you-can-eat buffet without knowing where to start. How do you pick what's worth your time and money in a world overflowing with choices?

Enter online movie reviews. In the digital age, online movie reviews play a crucial role in guiding viewers' choices. Among these platforms is Rotten Tomatoes [3].

Imagine trying to sift through endless streams of reviews, each with its own set of stars, helpful votes, pictures, and passionate dissertations on why a particular movie "changed [their] life" or how it was "two hours [they'll] never get back." Consequently, this overload of data may leave the average user lost, confused, and even frustrated.

Fear not, our project aims to develop a nuanced and multi-dimensional approach to analyze these reviews, leveraging a combination of factors to identify and recommend the best movies based on Rotten Tomatoes [3]. We're taking all those numbers, comments, pictures, and plotting them into an easy-to-understand graph. One can think of us as a personal movie concierge, guiding you to your next favorite film with just a few clicks.

There is no need for endless scrolling and movie night duds. Just you, a great movie, and perhaps a bowl of popcorn. Welcome to the future of movie selection, where every night can be the perfect movie night.

Research Question: How can we analyze and integrate diverse review metrics from Rotten Tomatoes to identify and recommend the best movies?

Datasets

1. rotten_tomatoes_movies.csv from Kaggle [3]

This dataset comprises records for movies listed on Rotten Tomatoes, extensive data corresponding to each movie.

Format

- **id:** A unique identifier for the movie.
- **title:** The movie's title.
- **audienceScore**: The audience score on Rotten Tomatoes.
- **tomatoMeter:** The Tomatometer score, representing the percentage of approved critic reviews.
- **rating:** The movie's MPAA rating (e.g., PG-13, R).
- **releaseDateTheaters:** The date the movie was released in theaters.
- **releaseDateStreaming:** The date the movie became available for streaming.
- **runtimeMinutes:** The duration of the movie in minutes.

- **genre:** The genre(s) the movie belongs to.
- **director:** The director(s) of the movie.
- **boxOffice:** The movie's revenue at the box office.

Fields Utilized: For our analysis, we focused on the id, title, rating, and genre fields. These were selected to examine trends and influences of movie genres and ratings on audience and critical reception.

2. rotten_tomatoes_movie_reviews.csv from Kaggle [3]

This dataset contains records for critic reviews of movies on Rotten Tomatoes, comprehensive overview of each review.

Format

- **id:** The movie's unique identifier, linking the review to its corresponding movie.
- **reviewId:** A unique identifier for the review.
- **creationDate:** The date the review was published.
- **criticName:** The name of the critic who wrote the review.
- **isTopCritic:** A boolean indicating whether the critic is considered a top critic by Rotten Tomatoes.
- **originalScore:** The score given by the critic, often in the format of a letter grade or numerical value.
- **reviewState:** The status of the review (e.g., fresh or rotten).
- **publicationName:** The name of the publication in which the review appeared.
- **reviewText:** The full text of the review.
- **scoreSentiment:** A sentiment analysis of the review score, categorizing it as positive, negative, or neutral.
- **reviewUrl:** The URL to the full review on the Rotten Tomatoes website.

Fields Utilized: Our project utilized the id and reviewId fields from this dataset to link critic reviews to their respective movies and aggregate review data. This focus allowed us to explore the relationship between the volume of critic reviews and movies' genres and ratings.

Computational Overview

Our project's computational framework is designed to analyze Rotten Tomatoes movie data to offer personalized movie recommendations.

1. **User Preferences Collection:** We first prompt users to specify movies and genres they prefer. This step is critical for tailoring the recommendation system to individual tastes, leveraging the `set_user_preferences()` function to collect this data.
2. **Graph Weight Assignments:** graphs play an indispensable role in representing the complex relationships between movies and their reviews, as well as capturing the connections based on genre similarities and audience or critic review scores.
    - **Weight Between Review and Movie (w1):** Represents the weighted average review score for each movie. This is calculated by aggregating critic review scores, highlighting the movie's reception among critics.
    - **Weight Between Movies (w2):** Indicates genre similarity between movies. It's computed based on the proportion of shared genres to total genres between pairs of movies, reflecting thematic closeness.

3. **Option 1: List of Recommended Movies** Beyond the initially preferred selection, this option ranks the top 10 movie recommendations. The ranking methodology integrates two critical weighting factors: w1, which evaluates the strength and relevance of review-based connections between movies, and w2, which assesses the degree of similarity between movies based on their genre characteristics. This dual-criteria approach ensures a well-rounded and tailored recommendation list that aligns with the viewer's preferences and interests.

4. **Option 2: Graph Centering on Preferred Films:** Using the preferred movies as a pivot, we connect every other movie in the graph to these selected movies using the weights (w1 for review-movie relationships and w2 for movie-movie genre similarity).

   We call `visualize_graph` to produce the graph.

5. **Option 3: Quadrant Analysis for Recommendations:** For visualization, movies positioned in the top right quadrant signify high similarity to preferences and positive reception, prioritizing them in our recommendations. This means that users are directed towards films that are both highly rated and aligned with their interests. This visualization highlights the balance between high ratings and thematic alignment, providing users with a nuanced understanding of recommended films. We employ a quadrant system where:

   - The y-axis represents overall similarity to the user's preferred movies/genres, calculated as a blend of 50% w1 and 50% w2.
   - The x-axis represents the average review score, determining a movie's quality based on critic ratings.

   We call `plot_movie_recommendations` to produce the graph.

6. **Threshold for Number of Reviews:** To ensure recommendations are based on a substantial amount of feedback, we filter out movies with fewer reviews than a specified threshold. This is aimed at reducing any biases stemming from a low count of reviews.

In our project, we leverage the CSV library to take in movie and review data, constructing a graph with movies and reviews as vertices, facilitated by our custom `WeightedGraph` class. [1] User preferences are captured through `get_favourite_movie` and `get_favourite_genres` functions, allowing us to personalize the graph around their interests. We further refine this graph using the `build_with_new_vertex_fraction` function to integrate review scores and genre similarities, assigning weights that reflect these aspects. Concurrently, the `add_vertex_to_simplified_graph` function applies a threshold-based filtering to focus the analysis on the most relevant movies. The recommendation process culminates in `print_recommended_movies` and `display_recommendations`, which translate the graph's insights into movie suggestions, either printed or through interactive visualization. This process harnesses Python's CSV and IO libraries, alongside our algorithms and data structures, to offer tailored film recommendations based on Rotten Tomatoes metrics. [1][2] Our computational framework thus not only addresses our research question but also pioneers a method for film recommendation, demonstrating the effectiveness of computational strategies in tackling real-world challenges.

The libraries pandas, NetworkX, and Plotly are harnessed to manage data manipulation, graph construction and analysis, and interactive data visualization, respectively. Pandas provides a robust framework for organizing and manipulating tabular data into structured DataFrames, enabling efficient data operations essential for analyzing movie and review information. [5] NetworkX facilitates the modeling of complex relationships between movies, reviews, and their attributes through versatile graph construction and analysis tools, preparing the data for visualization by determining node positions and edge connections. [4] Plotly, known for its ability to generate dynamic, interactive visualizations, brings the analyzed data to life, allowing users to explore movie relationships through customizable and aesthetically pleasing graphical representations. [6] The work done between pandas, NetworkX, and Plotly equips the project with a comprehensive toolkit for a deep and insightful exploration of movie data, from initial data structuring and analysis to the final presentation in an interactive and user-friendly format.

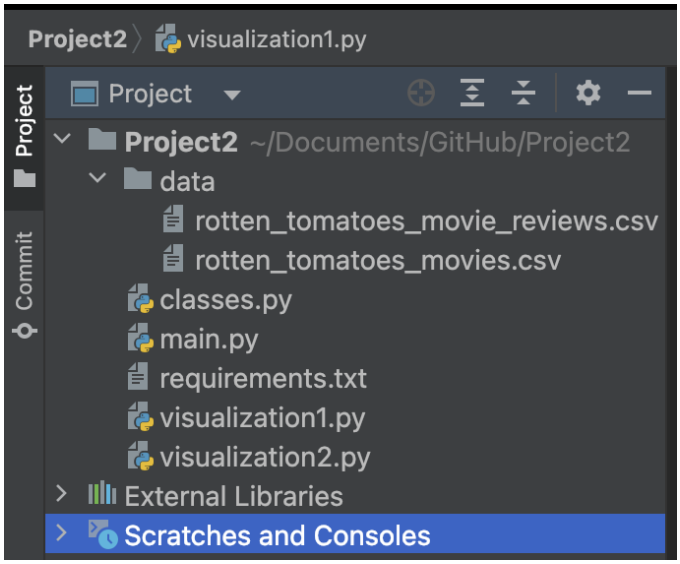
Tkinter plays a pivotal role in enhancing user interaction by enabling the collection of user preferences through a graphical user interface (GUI). By utilizing Tkinter's capabilities, the project prompts users to input their favorite movie and genres in a more engaging and accessible manner than command-line interfaces instead of simply a console. This GUI approach not only streamlines the process of gathering essential input for personalizing movie recommendations but also significantly improves the overall user experience. [7]

Instructions

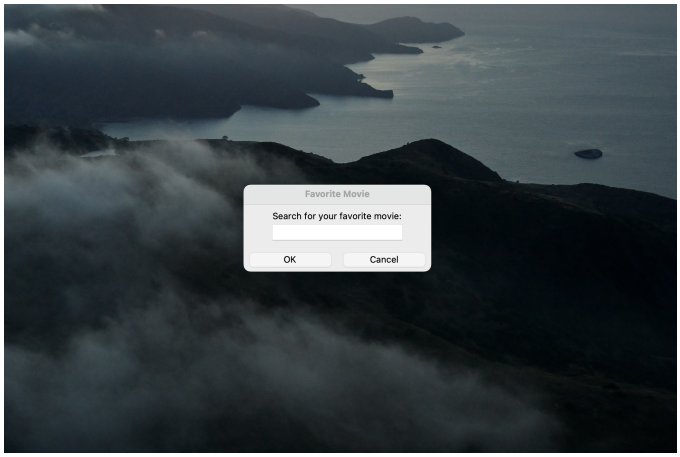
Please follow these instructions carefully to ensure you can run our program effectively:

1. Install all necessary Python libraries specified in the `requirements.txt` file.
2. The datasets `rotten_tomatoes_movies.csv` and `rotten_tomatoes_movie_reviews.csv` are essential for our analysis. Due to file size constraints, these datasets have been compressed into a ZIP file and are not sent directly with this report. You can obtain the ZIP file shared via `https://send.utoronto.ca`. Please use the course email address, `csc111-2023-01@cs.toronto.edu`, to access the file. Once downloaded, extract the CSV files to a known directory on your computer. They were sent by Reena Obmina (`rm.obmina@mail.utoronto.ca`).

   Claim ID: QJtKp4n3ZheMT6iW Claim Passcode: Xt9XE8HF45RqQ4i8
3. Ensure that the remaining Python script files for running the project (found on MarkUs for download) are placed in the same directory as the extracted dataset files.
4. The directory should look like this if done correctly.



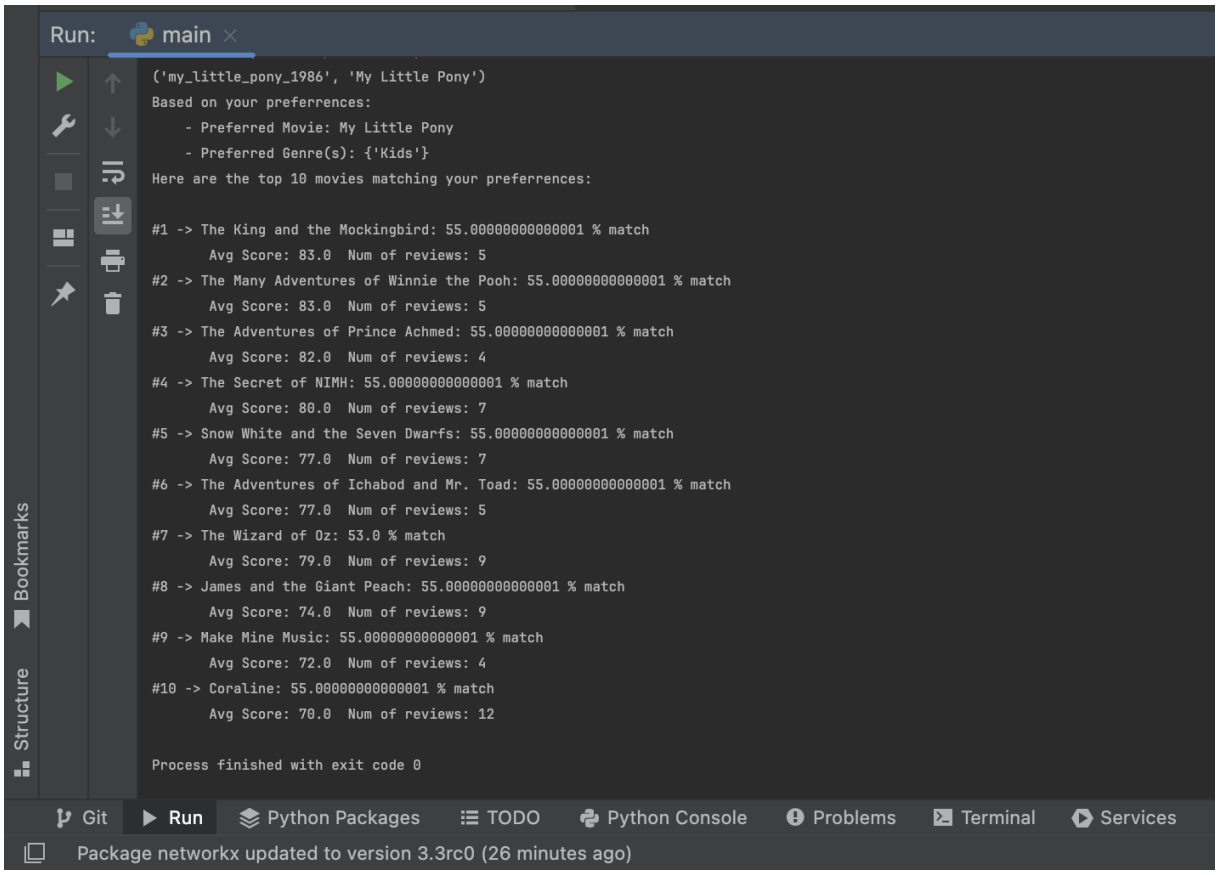
5. Navigate to the directory containing your project files and execute the main.py program file by first clicking Run 'main' -> then follow any on-screen instructions to interact with the program and receive movie recommendations via graph.
6. A screen like this should appear after following the above.



Visualizations

Default Print

This option is designated as the primary visualization choice and becomes accessible through the console upon selecting option 1. It thoughtfully curates and displays a list of the top 10 movies, meticulously ranked based on their similarity scores — with the most similar movie prominently positioned at the top and the least similar at the bottom. Additionally, for enhanced personalization and context, the console presentation includes the user's preferred movie and genres, providing a tailored reference point throughout the exploration process.

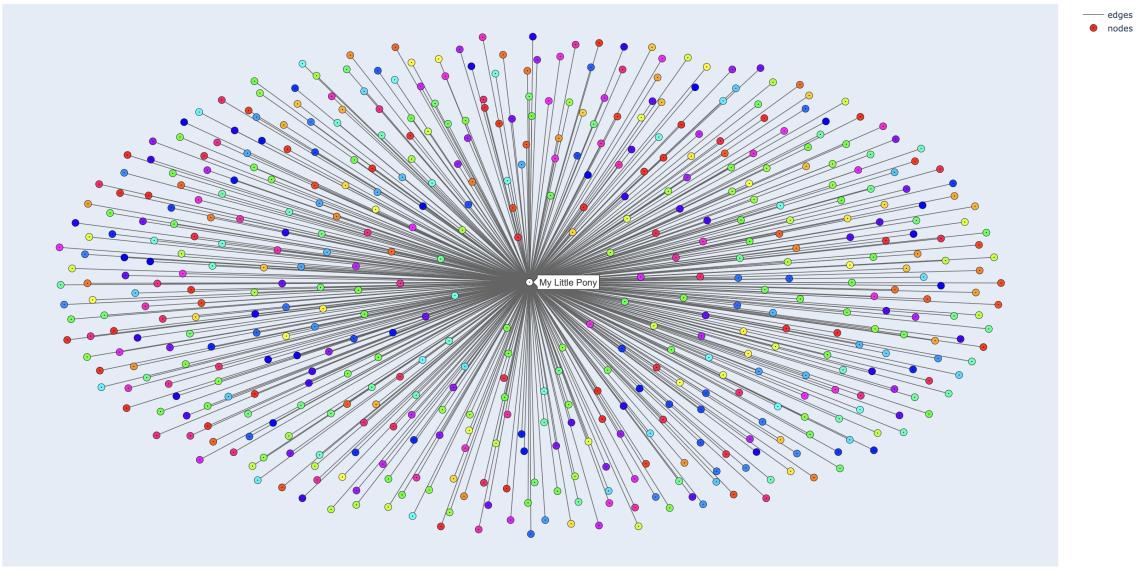


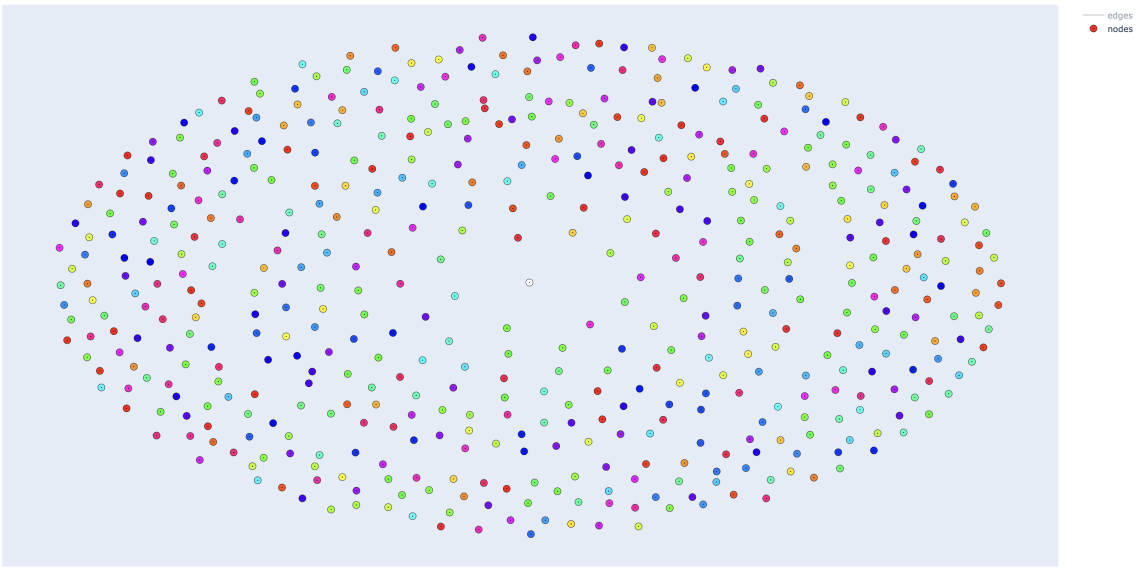
visualization1.py

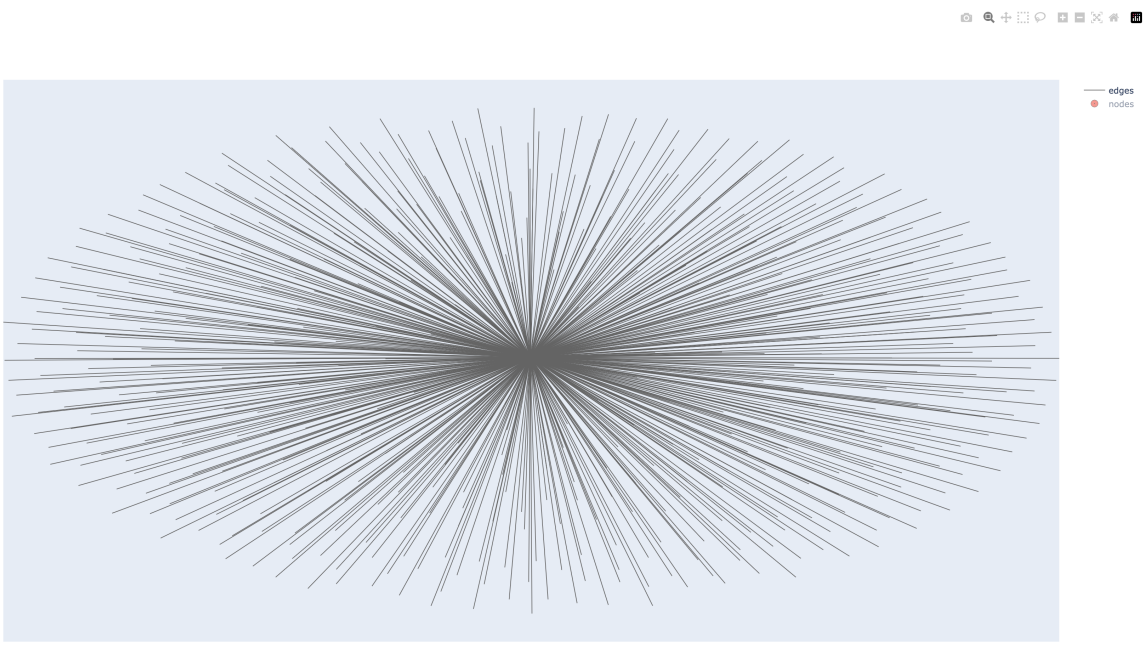
This module serves as the first visualization option and is designed to provide an interactive graph representation. We obtain this graph by choosing 2 in the prompt screen. It offers a dynamic way to explore connections within the data, enhanced by the inclusion of two critical interactive features. These features, accessible via buttons located in the top right corner of the interface, allow users to toggle the visibility of nodes and edges within the graph:

- **Nodes Button:** Activating this button will toggle the visibility of the graph's vertices. This feature allows users to focus on the relationships between items, represented by the edges, without the distraction of individual nodes.
- **Edges Button:** Clicking on the "Edges" button toggles the visibility of the connections between nodes. This option is invaluable for examining data points in isolation, free from the overlay of their interconnections.

These interactive capabilities are designed to enhance the user's ability to customize their analytical view, facilitating a deeper understanding of the underlying data structure by isolating specific components of the graph.

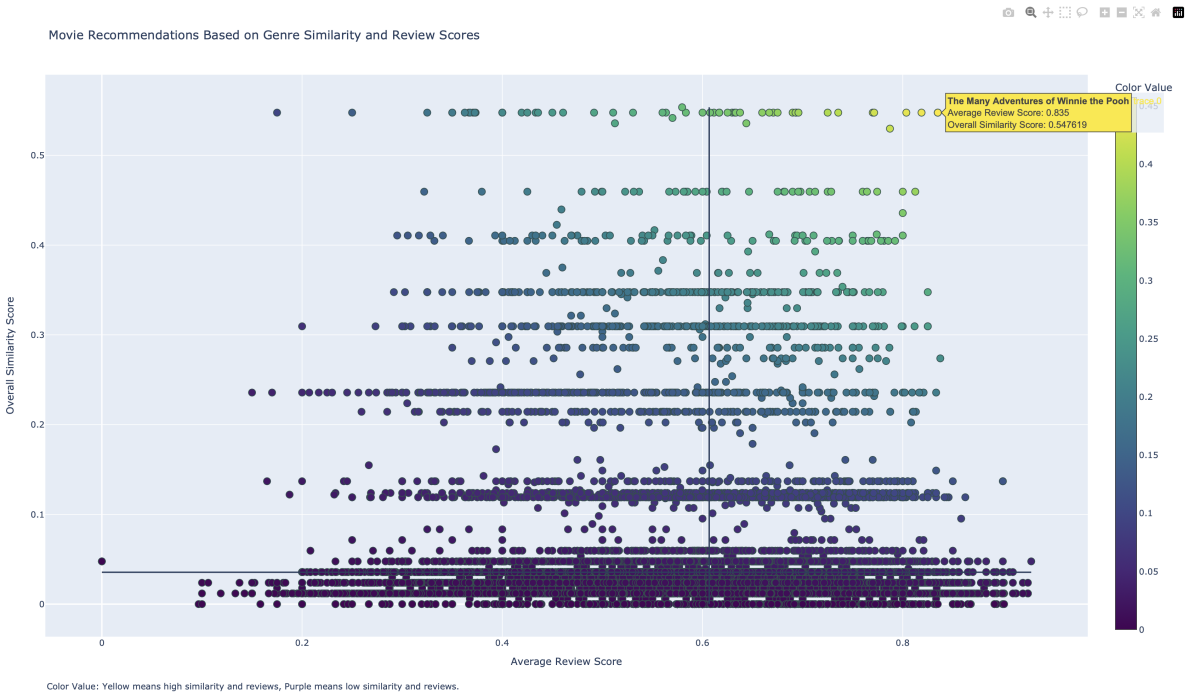

edges
nodes
My Little Pony


edges
nodes



visualization2.py

This module represents the second visualization option, crafted to offer a sophisticated method for ranking recommended movies. We obtain this graph by choosing 3 in the prompt screen. It utilizes both the Overall Similarity Score and Average Review Score to organize recommendations. Featuring a quadrant layout alongside a color value legend, this design intuitively guides users through the recommendations: movies with the most yellow hues are highly recommended, while those with deeper shades of purple are less recommended. This color-coded approach simplifies the process of identifying top movie recommendations, ensuring users can easily discern the most and least suggested options at a glance.



Improvisations

Our initial project proposal focused on leveraging Amazon's vast repository of movie reviews. We planned to create a graph-based analysis tool that integrates diverse review metrics to recommend the best movies.

The decision to change our analysis from Amazon to Rotten Tomatoes was influenced by several factors:

1. **Relevance** Rotten Tomatoes is specifically tailored to movies and their reviews, providing a richer and much more contemporary dataset for our analysis.
2. **Data Availability and Quality:** The Rotten Tomatoes dataset offers comprehensive review metrics, including genres, which were not available in Amazon's dataset.

With the shift to Rotten Tomatoes, our computational plan underwent several modifications:

1. **Dataset Processing:** We adapted our data processing scripts to accommodate the structure of the Rotten Tomatoes dataset, extracting relevant features for our analysis.
2. **Graph-Based Analysis:** We revised our graph construction approach to include both movie-to-movie and review-to-movie relationships, incorporating genre similarity and review scores as edge weights. This involved the introduction of a dual-weight system for the edges in our graph, where:
   - *Weight 1 (w1)* represents the weighted average review score for each movie, reflecting audience sentiment.
   - *Weight 2 (w2)* measures the genre similarity between movies, providing insight into thematic connections.

   These modifications allow for a more significant analysis by considering both the quality of reviews and the thematic links between films.
3. **Transition from Goodness Score to Multi-dimensional Analysis:** Originally, our plan focused on developing a "goodness" score for movies, based primarily on a combination of helpful votes and overall star ratings. However, we shifted towards a framework that incorporates both the weighted average review score (w1) and the genre similarity score (w2). This change was motivated by the realization that movie recommendation complexity requires a broader set of metrics for a more comprehensive understanding. The "goodness" score was replaced with an approach that evaluates movies based on:
   - *Review Quality* (w1), calculated as the weighted average of review scores, capturing the general sentiment towards the movie.
   - *Thematic Similarity* (w2), a measure of how closely the genres of two movies align, facilitating recommendations based on thematic content preferences.

Discussion

The primary objective of our project was to leverage the vast array of review data from Rotten Tomatoes to recommend films that align with user preferences, effectively answering our research question: "How can we analyze and integrate diverse review metrics from Rotten Tomatoes to identify and recommend the best movies?" Through our computational exploration, we developed a system that not only considers review scores but also factors in genre similarity, aiming to provide a holistic movie recommendation system.

Our approach to representing movies and their interconnected relationships through a graph significantly contributed to achieving this goal. By assigning weights to both movie-to-review and movie-to-movie edges based on review scores and genre similarity, respectively, we created a model that reflects the multifaceted nature of movie appreciation. However, the complexity of accurately reflecting these weights in our visualizations presented a considerable challenge, particularly in `visualization1.py`, where the representation of vertices did not convey the edge/connection relationships as we had hoped when it came to visualizing the first graph.

Our initial vision involved a graph where movies arrayed around a chosen favorite indicated their recommendation strength through proximity; closer movies suggested a higher relevance based on review scores and genre similarity.

Achieving this in a manner that was both analytically meaningful and visually intuitive proved challenging. The task demanded a level of customization beyond the default capabilities of the visualization libraries at our disposal.

The second visualization, which utilizes a quadrant system, represents a significant step forward in how users can interact with and interpret movie recommendations. By categorizing movies based on their similarity to user preferences and overall review scores, this approach allows for a multifaceted exploration of potential film choices. This quadrant system is not just a static representation; it invites users to engage with the data dynamically, offering a more nuanced understanding of how each movie aligns with their personal tastes and the collective critical reception.

This method stands out by providing a clear, visual delineation of movies that are highly recommended (top-right quadrant) versus those that might be of lesser interest (bottom-left quadrant), based on the intersection of user preference similarity and review score metrics. Such a categorization enriches the decision-making process, allowing users to quickly identify films that strike the best balance between popularity (as evidenced by review scores) and personal relevance (as indicated by similarity to chosen preferences).

Despite these efforts, our project was not without its shortcomings. The limitations we encountered with our datasets, particularly in the granularity of demographic data and the depth of review content, constrained our ability to offer even more personalized recommendations. Moreover, the challenges we faced in visualizing weighted vertices underscore the ongoing need for more sophisticated graph visualization tools that can handle complex, weighted relationships with greater ease and clarity.

Looking ahead, there are several avenues for further exploration and improvement of our project. Developing more advanced algorithms for weighting and visualizing relationships within our movie recommendation graph could enhance the accuracy and user-friendliness of our system. Additionally, incorporating user profiles based on their review history and preferences could lead to more personalized recommendations, while expanding our data sources to include social media sentiment or box office performance data could provide a better view of a film's overall reception.

In conclusion, our project represents a step forward in the application of computational techniques to the domain of movie recommendations. While we successfully developed a system that leverages Rotten Tomatoes review metrics to offer tailored film suggestions, the journey highlighted significant challenges and opportunities for further refinement. Our exploration underscores the importance of sophisticated data analysis and visualization tools in extracting meaningful insights from complex datasets, and it sets the stage for future work that could further revolutionize how we discover and interact with film recommendations.

References

[1] "csv — CSV File Reading and Writing." csv. `https://docs.python.org/3.12/library/csv.html` (accessed April 1, 2024).

[2] "io — Core tools for working with streams." io. `https://docs.python.org/3.12/library/io.html` (accessed April 1, 2024).

[3] Andrea Villa, "Massive Rotten Tomatoes Movies & Reviews." (2023). Distributed by Andrea Villa. `https://www.kaggle.com/datasets/andrezaza/clapper-massive-rotten-tomatoes-movies-and-reviews/data` (accessed April 1, 2024).

[4] "NetworkX Documentation." NetworkX. `https://networkx.org` (accessed April 1, 2024).

[5] "pandas: powerful Python data analysis toolkit." pandas. `https://pandas.pydata.org` (accessed April 1, 2024).

[6] "Plotly Python Graphing Library." Plotly. `https://plotly.com/python/` (accessed April 1, 2024).

[7] "tkinter — Python interface to Tcl/Tk." tkinter. `https://docs.python.org/3.12/library/tkinter.html` (accessed April 1, 2024).