Dataset: Netflix TV Shows and Movies
https://www.kaggle.com/datasets/victorsoeiro/netflix-tv-shows-and-movies/data?select=titles.csv

For this final project, I decided to use a dataset from Kaggle that provided information on numerous Netflix TV Shows and Movies, displaying the actual title, actors and actresses starring in the show/movie, a brief description of the film, the release year, genres, etc. When first starting this project, I realized that the Kaggle link I got my dataset from provided me with 2 .csv files, one containing the titles of the films (which were given unique movie_ids) and another one containing the actors/actresses for those unique ids. As I was looking to connect the titles with the actors/actresses, I began my project by merging the two CSV files into one, in order to make the rust implementation a bit simpler, combining the titles.csv and credits.csv files from the Kaggle into one CSV file called merged_movies.csv.

My rust implementation builds a graph-based model to analyze and understand connections between movies through their shared actors. By creating a graph where each vertex represents a movie and each edge signifies that the two movies share at least one actor, the program can calculate the similarity and dissimilarity between movies based on the number of shared actors.

The key functionalities of my project are as follows:

1. Graph Construction: My project starts by reading a CSV file containing the titles of each film and actor names, and constructs a graph based on the aforementioned guidelines. Each movie becomes a node in the graph, and an edge between two movies exists only if they share at least one actor.
2. Similarity Calculation: In order to calculate similarity, I made use of the Jaccard similarity index. In doing so, my program computes the similarity between every pair of movies. The similarity was measured by determining the ratio of the number of shared actors to the total unique actors across both movies. A similarity index of 1 indicates that both films had an identical cast while a similarity index of 0 indicates that the films shared none of their cast.
3. Identifying Extremes as Outputs: This project then identifies and reports the pairs of movies that are the most similar (with the highest similarity score) and the least similar (with the lowest similarity score) and returns them as an output.

To run this project, one must begin by downloading the project to their local machine, then run the “cargo build” command to build the project. Then, they can proceed by running the code by using the “cargo run” command or the “cargo run --release” command (for faster results). To run the tests, one can simply run the “cargo test” command. This command checks the correctness of the movie and actor creation, graph construction, and similarity calculation portions of our code.

A sample output for my project is as follows:

- Most Similar Movies: (Movie { title: "Trailer Park Boys: Out of the Park: USA" }, Movie { title: "Trailer Park Boys: Live at the North Pole" }) with similarity score: 1
- Least Similar Movies: (Movie { title: "How It Ends" }, Movie { title: "Freud" }) with dissimilarity score: 0

This output shows that in this run of our code (as numerous shows/movies share identical casts, whether its due to sequels or specific shows hosted by one star, or share none of their cast) the most similar movies were *Trailer Park Boys: Out of the Park: USA* and *Trailer Park Boys: Live at the North Pole*, with a similarity score of 1. This shows that they shared an identical cast, which is logically sound as their titles highlight the fact that they are most likely part of the same series. It also shows that the least similar movies were *How It Ends* and *Freud*, with a similarity score of 0, meaning that they shared none of their cast.

Overall, this project provides a comprehensive analysis tool for understanding the relationships between movies based on their casts. It is especially useful for data analysts and movie enthusiasts interested in uncovering hidden patterns in casting across the film industry. As a movie enthusiast myself, I chose this dataset for this exact reason and enjoyed implementing this project very much.