Block-Cluster: Visualizing Movie Success

Sana Ajani, Sahaj Bhatt, Manoaj Kandiakounder, Matthew Owens, Rahul Patel, Dhruv Sagar

Georgia Tech

Summary "The Synopsis"

Today, the way we perceive trends in the movie industry is largely impacted by social media and subjective reviews. Ironically, for such a large industry, there are no commonly available tools for **visualizing historical performance**. Our goal is to provide a set of objective criteria to enable the user to obtain a better understanding of trends in the movie industry. Our project deliverable is a dynamic tool to discover relationships between movies and visualize how different attributes influence a movie's success. The final visualizations display the success metrics of each movie and allow the user to interactively filter movies and choose attributes to cluster on.

Our Approach "Director's Cut"

1. Algorithm

Initially, the app shows a conglomerate visualization. The size of the node is proportional to the movie's international revenue. The deeper the color, the more award wins and nominations a movie has. This helps the user gain insights on the relationship of a movie's "success" with its position in a cluster. Next, we give the user the power to either **filter the visible movies into a smaller subset** of their choice or cluster them based on available movie attributes. We utilize a **K-means clustering model** based on multiple experimental results for this visualization. For filtering, we provide the user with different visualizations based on their selected attributes.

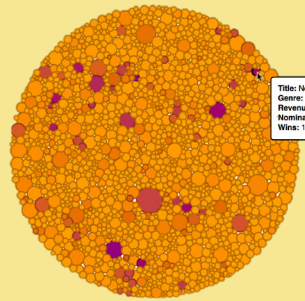
2. Visualization

As shown below, we created a calendar visualization for release date filtering, and a force-directed bubble chart for genre filtering. This new approach enables the user to choose from a variety of clusters and filters, allowing for a **high level of customization and insight** into performance trends. We believe this fills the void for a visualization tool that can help a user uncover unique movie industry insights.

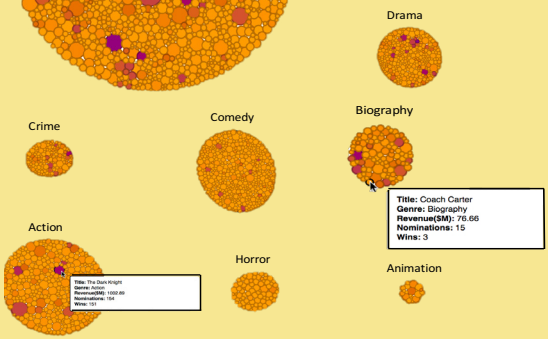
Experiment "The Plot"

We used a PCA technique to transform the data with 21 binary features into a linear combination of the genre feature and ran K-means on the transformed data. In order to evaluate this approach and choose the best number of clusters, we used quantitative metrics, such as Silhouette score, along with how well the resulting clusters fit the user experience. Furthermore, we evaluated the effectiveness and intuitiveness of our potential visualizations on a subset of users by soliciting feedback throughout the development process. Since we use a **distinct combination of attributes** to compare movies, our method is **more objective** compared to current user review driven success metrics.

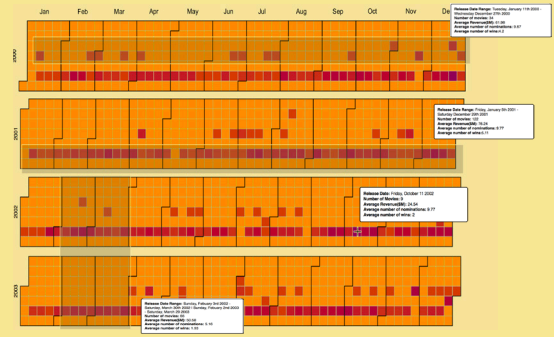
Examples of Filtering "Eye Candy"



User filters by "genre"



User filters by "release date"



Data "Cast & Crew"

We accessed IMDB datasets from an AWS S3 bucket to retrieve information such as movie titles, genres, runtime, ratings, and principal cast. We used Python scripts for API handling, and web-scraping with the BeautifulSoup library to obtain date of release, award nominations, and box office impact. We stored this in a locally created SQLite database. We limited our dataset to 2000 movies released within the 21st century, randomly choosing from all 22 different IMDB genres. This resulted in a 1.5 GB sized file, that we then cleaned via OpenRefine to convert into one consistent format.