My IMDB Analytics Dashboard Application

- **Introduction**

This programming project has been to develop a visualization tool which can be used to gain broader insights about movies through information available from IMDB. The tool is intended to help users considering watch recommendations, hobbyists looking to relate their experience with broader insights, and those questioning what the potential will be for a title. A tool like this can also help its users make more objective choices when engaging with IMDB. By consulting with the extent to which votes are skewed to the extreme values of zero or ten, an impression can be given of the volatility a title may have. By relating existing user reviews to the given title and its similarity with other closely related titles, as well as those which have achieved similar rating and engagement, a deeper perspective for a user when proceeding to write thier own review is aided. Perhaps with a broader sense of where a title stacks up already, some review that exists already, or whether a title has been subject to heightened volatility a user could reduce the volatility of their own analyses and improve the platform as a whole.

The current core dataset is the IMDB Non-Commercial Datasets. A series of datasets provided for free by IMDB to allow for hobbyist analysis and historic interpretations. A link to the webpage providing the datasets and their documentation will be provided here. Those datasets are provided in tabular form, via tsv format stored in gz archive files. Although these datasets are very robust, much more has successfully been made from them. As an extension to the non-commercial dataset, it became necessary to source data dynamically from IMDB for a distribution breakdown of existing user votes on titles. In addition, IMDB lacks financial data in their free datasets like budgets and gross, or any inclusion of existing user review data. This can be accessed directly via pages associated with the title, so functionality has been icluded to incorporate them.

The result of the project is a functional analytics dashboard which is served to users as a web application. It is intended to be accessed through a browser, and to be used in conjunction with the IMDB website. Overall I am very satisfied with the applications's performance; there was some concern for this as the size of the data used is fairly significant and the hardware I have used is somewhat modest. Nearly all of the originally proposed functionality for the project proved feasible by the end. Some of which may have been a matter of coincidence, but is largely due to the thorough development of the libraries used and wide availability of helpful information online surrounding their use. In the end, the project has allowed me to gain experience for using visual analysis design and

several new information processing and presentation libraries for the python programming language, which I had never used before.

- **Dataset**

The quality of the dataset provided by IMDB leaves something to be desired, but the ultimate goal of this project requires accommodating these shortcomings – which I will discuss here. The process of analyzing the data quality and cleanliness has led to me reevaluating what data is to be included a couple of times so far. Subjectively, the items of the table serve an important purpose even where they are very incomplete. Objectively though, more than half of the items here have few populated attributes and/or have experienced no engagement from users. This limits how they can have a report generated for them, and led me to an internal debate about their inclusion. Finally, I was able to work in functionality to respond to requests for titles which either would not qualify as a movie title or could not have a report generated due to unavailability of rating/votes information. This led to visualizations generated from two effective versions of the dataset. A larger, more comprehensive version used to influence the project's historic view and limited, able to generate a report subset; both comprised of the same attributes.

Preparing the dataset I have submitted required integrating two tables from those provided by IMDB. These were specifically "title.basics.tsv" and "title.ratings.tsv". Other tables provided in the non-commercial set bore little relevance to the goals of the project, and were excluded. Also, some attributes originally in the basics table are not included in the submission dataset, but were used to preemptively filter entries or were not used with movie titles. Specifically, "original title" proved redundant, "end year" is only for TV series, and "is adult" was excluded after the items which qualified as adult were filtered out. Additional wrangling, altering attributes to have a correct type and imposing replacement in some cases, was also necessary. Interestingly, the ratings and number of votes attribute for the submission set is actually half derived. Because of IMDB's method of excluding titles from holding entries in the ratings set where they have yet to receive five or more votes. This is why items can be seen to hold either zero, or five or more votes in the case that rating is null or non-null respectively.

Rather than attempt to use the fully comprehensive tabular data in practice, additional wrangling yielded positive performance results for the many visualizations the dashboard provides. This took the form of pivoting and aggregating elements into four additional small tables to construct the bulk of the historic view page. In order to further complete reports for titles it was

necessary to source some information dynamically, where it was not directly included in the non-commercial sets provided by IMDB. Through existing python libraries like BeautifulSoup and urllib, these additional attributes are included just like other features, although no mass scraping of the IMDB site is used for the project.

The submission dataset is provided as a tsv file. The choice of tab separation keeps with the format selected by IMDB, and is specifically necessary for the title and genres attributes as they can include commas – or are often a comma separated list with genres. Even though the dataset previously submitted for the project has developed beyond a single table, the majority of attributes provided in the original dataset submission remain. The only feature since last submission to be excluded is the categorical attribute "type". All static datasets used are tabular data. Items of these tables hold the following attributes; broken down by title, attribute type, and semantic description.

tconst	type	title	year	runtime	genres	rating	votes
tt0000009	movie	Miss Jerry	1894	45	Romance	5.4	212
tt0000147	movie	The Corbett-Fitzsimm…	1897	100	Documentary,News,…	5.2	517

Original Dataset Submission Sample

- **tconst (categorical)** - alphanumeric unique identifier of the title
- ~~**type (categorical)** – the type/format of the title (e.g. movie, tvmovie)~~
- **title (categorical)** – the more popular title / the title used by the filmmakers on promotional materials at the point of release
- **year (ordinal)** – represents the release year of a title. In the case of TV Series, it is the series start year
- **runtime (quantitative)** – primary runtime of the title, in minutes
- **genres (categorical)** – includes up to three genres associated with the title
- **rating (quantitative)** – weighted average of all the individual user ratings
- **votes (quantitative)** – number of votes the title has received

In addition to these I sought out further data dynamically, such as percentage breakdown of the vote histogram, which is a distribution of quantitative data for a given title, and nominal information about the title presented as a heading for each search.

- **Tasks**

The project goal has been to develop a visualization tool that can be used to gain insights about movies using a broader scope than what can be learned from just examining an individual page or trying simultaneously to consider several of the many hundreds of thousands of individual movie pages provided on IMDB. It aims to do this through a web application dashboard which provides a collection of visualizations organized between two formats or "views". The first is the Historic view, used to greet users each time the application is accessed. The second are Title views, which users can access by searching on the dashboard where they have a specific title in mind.
Some descriptions of tasks available to users of the tool include:

Get broad information (via visualization) about movies based on some limited identifiers, like genre, current rating, year, etc.

Discern some of the volatility of a title (this is a poorly addressed issue for IMDB).

Discover from a sample of closely related titles and relate them to a currently searched one.

Observe some existing review response in conjunction with the visual information presented.

The most immediate access for the user to this tool is through its historic view, which offers insight based on broad analysis of the core dataset but does not regularly update. By providing this view of the data, it will be better ensured that the user has a consistent "overall" interpretation of IMDB use and the characteristics of the movies presented there. From here the option to look up a title via its 'tconst' (the attribute used to uniquely identify titles on IMDB) cues the construction of a narrower trends view for the title. The intention here is to highlight the title relative to those of the same genre, close temporal proximity, and similar engagement and rating. Several visualizations are presented to give the user an overview of the title's current standing. Moving deeper, because the user is provided with some existing IMDB response they may then choose to proceed with their own engagement; whether they cast a vote, write a review, try to watch something else, or ignore the title further.

Using Munzner's terminology for abstracted tasks, at the high level the hope of the project will be to offer users discovery and enjoyment through data sourced from IMDB. Users may ideally have the goal to produce their own input

as a result of the application, but this would be directed to IMDB rather than involve or alter the dataset used here.

The two major view formats of the project will provide engagement to the user for either lookup or explore style searching. This also distinguishes their particular "query" scope. The Historic view focuses on the summary of the records, whereas the Title view serves to directly identify and compare them. The two views also differ in what targets are used to support their actions. The Historic view makes targets of the broader trends, similarities, and (to some extent) the collective shape of titles – where we examine them under a lower granularity interpretation. This means analyzing the whole of the dataset, or at most subdividing the whole along only particular categorical attributes. Title views differ their targets by focusing on correlations and similarities of much more heavily filtered subsets of the data and the distribution of a title's votes attribute.

- **Visual Design – Historic View**

The Historic view supports the task abstraction by sticking to broad, low granularity depictions of the dataset. This is accomplished by relying consistently on larger scale groupings of the records, averages of attributes, wide temporal analysis, and distributions across the span of the data. How scantly populated the records are, which comprise the whole of the IMDB dataset, gets direct examination. The proposed domain of the project is more effectively supported by highlighting the presence of these subsets.

The Distribution of Global Ratings histogram and its annotations are a key feature of the view. It shows users that the tendency of ratings given exhibits normal behavior as an average, and that the population of movies can be explored in terms of the statistics which can be drawn from the histogram. In other areas of the application, the standard deviation of ratings is used as boundaries for what is considered a "local neighborhood" of a given title. The interactivity of the histogram is straight forward. Each bar of the histogram provides tooltip information to help the user observe the size of the bar, and associated rating, precisely. The histogram was not the original visualization choice for this particular realestate of the view, but conveys very important characteristics about IMDB and its movie population up to now. This may help lead users to understand more objectively how they might define what qualifies as "good" vs. "bad", which could not be accomplished through standard engagement with IMDB; this supports the goals of the project.

The Per Page Completion Breakdown sunburst & pies are a joint visualization showing the overall percentages and interrelation of incomplete attributes for

Historic View Example Screenshot

the movie pages of the database. The sunburst is a more unique and interactive idiom. The user may interact with the sunburst by clicking its regions, causing the chart to shift in order to illustrate the relative dimensions of cells specific to movies which either possess or do not possess its related attribute. For example, the population of titles which do not have any of the four major attributes presented is very small, yet the population of titles which have never had sufficient votes to possess a rating is greater than those which have. The intention of the pie charts is to provide a static illustration of the ratios of attribute completion, but the sunburst itself provides an interactivity that illustrates how attribute completion is related. There are a few visualizations which are very closely related to sunbursts which show data in a similar way, but ultimately the sunburst is the more standard choice with the plotly library and is far more engaging than alternatives like stacked bar charts.

The Yearly Averaged Votes line chart shows more than which years of movies happen to have more engagement. It shows an inherent bias which exists towards titles which released in close proximity to the site taking off, and can help users

to consider a goal of the exploration of years that have so far had less engagement, such as any decade prior to the 1980s. There may also be something of a feed-back loop illustrated here, where users seek out the "best" options, and thereby drive further attention to those same options, creating a bubble of exposure. Interactivity here relies on the tooltip and axis zooming. Where users want to explore a change in the trend of the line, they can select an area or just hover to read the exact value.

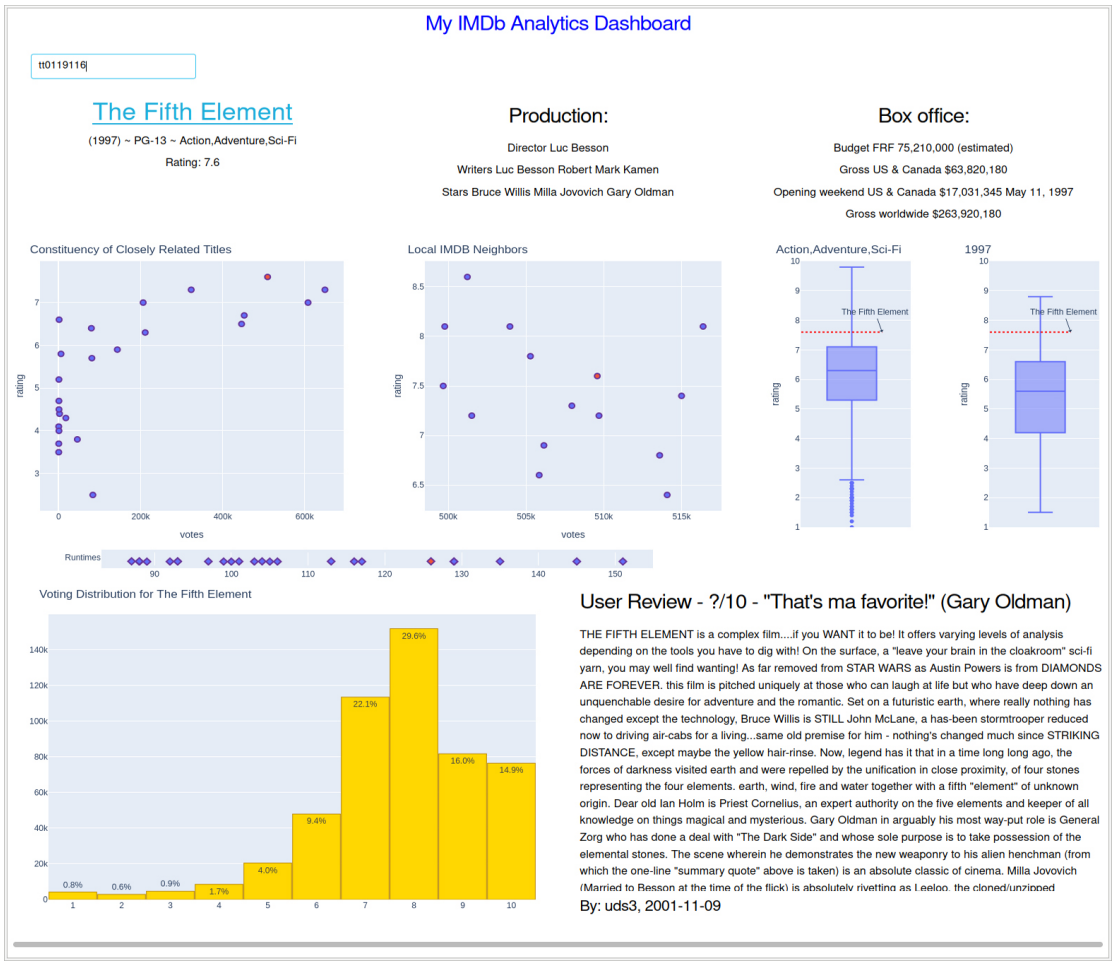
The Yearly Averaged Ratings line chart is of course similar to the previous chart. Users are still able to learn important information about the site from it though. User reactions to the earlier decades of movies are rocky and inconsistent, but call attention to them as a possibility of an unexplored area of the art form to be engaged with. There is also relatively surpising consistency to the trendline accross the rest of the decades leading up to most recent years, as well as a behavior of interpretations of the newest releases generaly starting high before becoming more even handed. These behaviors could not be illustrated without some form of trend line, making them a reasonable choice for looking at the data in this context.

The Votes & Rating Averaged by Genre scatter plot illustrates about twelve hundred different genres. The behavior shown by it seems very unique to the context of movie ratings, with more engagement influencing regression toward the mean and many extreme outliers. Other possibilities for the same aggregated data were considered due to overplotting, such as a heatmap, but this won out because of the interactibility of plotly graphs and the easy interpretability of scatters. A general region may be selected with the mouse to examine the plot more closely where a zoomed out view prevents distinctions, though this is still helped by the tooltip.

The Averaged Runtime by Gerne violin plot does a great job of making use of the space of the view, and reveals a good variety of statistics when interacted with - like the median runtime or its most extreme values. Users may not anticipate deviation when titles are aggregated by genre (one classmate of the course communicated this directly), but the variation is fairly exaggerated and justifies a very tall plot. Rather than sticking to a traditional violin plot, plotly customization allowed all genre marks to be kept visible so that users can use tooltips to directly identify which genres tend to have runtimes that they might prefer and which they may wish to try and avoid.

- **Visual Design – Title View**

A Title view is accessed via the search utility and focuses user interaction to a more confined subset of the titles in the dataset. It supports the task



Title View Example Screenshot

abstraction by providing individualized information like votes distribution, budget information, and relation to other titles included – those similar in key metrics, referred to as "constituents". Filtering down to see the relationship between the title and others in its respective genre or year, or to relate the title to others with close identities across multiple attributes serves the user goal of gaining insight which cannot be found solely through its IMDB page.

The Constituency of Closely Related Titles scatter plot serves a unique purpose for the view. It uses fairly particular querying of the dataset to share with the user a sample of titles, up to one hundred in size, of any other titles of the same genre and released within a breadth of seven years. This sampling cuts down on any possible overplotting. The default scaling of the graph is entirely subject to the sampling, which changes with each search of the same title – where there are sufficiently large numbers of "constituents". By providing this the plot directly supports tasks like considering watch alternatives or observing how a title stacks up visually. This is made easy thanks to linked highlighting used with this plot and others on this row of the view. Tooltip data is also more

elaborate for the plot, encouraging the user to hover over in order to see what similar titles have shown up here and make further searches accordingly.

The Local IMDB Neighborhood scatter plot provides a different take what is "similar" to the searched movie. Instead, so long as another title appears within a fairly close range for votes (several thousand) and rating (within a standard deviation) then the title may be selected via sampling to be shown as a member of the neighborhood. Linked highlighting is included to keep the searched title distinct, and tooltips allow the user to understand which specific movies exist within this region of the overall ratings by votes space of IMDB.

The Genre Comprehensive & Year Comprehensive box plots help to show the searched title in a different light by annotating its position within its respective year or for that genre as a whole. This helps make sure that the user observes the rating performance of a title without the two scatters restricting interpretation by taking a sample of limited size. Because these box plots show data along a wider scope, the user is able to draw more from the associated genre or year when taken from the historic view, which was confined to only an average. Through hovering, the user can see statistics like different percentiles for a box, and gets a clear sense of how the searched title compares to those percentiles.

The Runtime plot is a single axis plot based on the plotly boxplot which only shows the actual marks for the runtimes of the searched title and constituents. The title's constituents are used specifically because users may want to stick to the same genre for a movie they pick, but need to base a decision on more details than just how something is rated or how visible the title may be. This should be clear when tooltips are used to see which title belongs to each mark, and can aid in moving forward with a watch suggestion when considering time as a limiting factor.

The Voting Distribution for Title bar chart is particularly useful to have shown in immediate relation to other visualizations, rather than being effectively hidden behind an alternate page for the title where IMDB keeps them. It serves the task where a user may be considering how to vote on a title they have seen already, or to anticipate potential volitility for a title. It was important to source this information for the dashboard, because for whatever reason it is not included in the non-commercial data sets yet is one of the most critical pieces with which to analyze titles. The chart interacts in the standard way as other plotly bar charts. The y-axis represents direct count of ratings and is viewable through tooltip, although effort was made to label bars by their individual percentage of voting total to give those details at a glance. The overall shape of the distribution is its key value to the user, and a histogram is the standard method used to show this interpretation for the data type.

- **Results**

The project is written in the python programming language and uses the flask web framework to serve the dashboard, using the pandas and Dash by Plotly libraries to facilitate processing of the dataset and generation of the visualizations, respectively. Some data is also required from the IMDB website directly to produce Title views, which involves web scraping performed via the urllib, json, and BeautifulSoup4 libraries. These proved an invaluable portion of the overall functionality.

The pandas library serves to manipulate the tabular data provided through IMDB's datasets. It was used to perform the initial wrangling of the datasets to complete the dataset portion of the semester project, and is traditionally a part of the active processing involved to produce visualizations through Dash by Plotly. Plotly itself is of course one of the several popular graphing libraries. So Plotly contributed the entirety of the graphing functionality for the application. Graphing with Plotly and handling with pandas is combined under the functionality of the Dash library; also an offering from Plotly. Dash itself uses the Flask web framework to generate the underlying server, making it simpler for developers to create and deploy interactive dashboards. When a Dash application is initialized, it internally sets up a Flask server to handle HTTP requests and manage routing. This allows Dash to serve the web pages, handle user interactions, and update the application's state. Dash components, which are defined in python, are converted into HTML, CSS, and JavaScript and then served to the client by the Flask server.

The use case for the dashboard is simply that the user access its web address via a browser and have some movie in mind to look up. Lookup is accomplished using the associated 'tconst' value for the title, which can easily be found by looking to the url of any movie page on IMDB. From here, the user can engage with the information generated by exploring visualizations and making subsequent searches using a tconst from other titles presented in visualizations via hovertext. Where the user feels inclined to engage with the IMDB page of a title they have searched, they can always follow the link available at the top of the page.