Andy Snitgen

Professor Karen Jin

Comp574 – Applied Computing II

November 1, 2021

**Project 007**

***Data Collection***

**API Data Collection**

The data used in the 007 project primarily comes from Internet Movie Database and is accessed through their API. IMDb requires an email to get the necessary key. The program uses a key that is linked to my student email. While IMDb allows for 100 free API calls per day, the key in project007.ipynb will provide 5,000 calls per day until November 23rd, 2021. If the project requires more than cursory usage after that date, I will likely purchase another month. Their documentation warns against using multiple emails to obtain more keys, and thus free calls. In fact, it suggests that the practice could result in an IP ban, and I would like to avoid that. It is an important fact to consider because the code within project007.ipynb requires 54 to 58 API calls per successful run. That is the equivalent of only a single use per day.

The small variance in number of API calls is due to an algorithm that detects if the API contains box office data for No Time to Die. That is the newest James Bond movie, and it is currently in theaters. It opened in the UK on September 30th and in the USA on October 8th, 2021. This is interesting to note, because the API contains data for the international market, but returns only a blank string for the US. While I could have scrapped the data from another website, I noticed IMDb's own webpage has the data. So, I wrote a function called checker that ascertains whether that information has been placed in the API. If the checker function returns a false value, the program instead inserts the accurate data that I hardcoded into the program in place of the empty string. I chose to hardcode it instead of pulling it from a different source to maintain consistent methodology. In my experience, different sources can provide different information; I trust IMDb's data to be authentic and stuck with them throughout. Ultimately I placed priority on providing an accurate and useful database for comparisons.

Project 007 uses API calls in two places, both functions. One of them is the aforementioned checker function, but usage primarily comes from the data function. In fact, the checker just uses a shorter, more concise version of data's code. In essence, checker pulls data only for No Time to Die's missing values to confirm that are, indeed, still missing. Below I will provide a brief description of how the data function works.

*Data Function Process Flow*

Text

Description automatically generated

**Interpolated Data**

Some values used in the dataframe are not drawn directly from the API, instead they entirely interpolated. They are the best estimation I could produce given the concrete data the IMDb API provided. I chose to use approximations because it allows for more useful comparisons than would otherwise be viable. One example that has already been discussed is the case of a null value for IMDb user ratings. In that case, the only other relevant data that is accessible is the Metacritic aggregate score. With no other data points, the best approximation of a user rating is the critics. So, the Metacritic score was simply converted to the IMDb rating format and appended with an estimation tag. While this only affects the recent No Time to Die, I thought it prudent to develop an algorithm capable of handling null edge cases in case other movies are added in the future.

That single instance is not the only approximation in the program. There is another, much more common case of interpolated data found in the international revenue column. As discussed, when foreign box office data is unavailable or limited in scope, the international revenue will be equal or very close the US box office. Specifically, if the international revenue is within 1% variance of the US box office I deemed the data to be missing and replaced the value with the string 'No Data'. The missing data makes any comparison of total revenue between films invalid, so to compensate I devised an algorithm remedy the problem. It can be found in the tenth code block of the project007 Jupyter notebook. It contains to sections, headed by comments that are described with an abstract style of pseudocode below.

*Interpolating Worldwide Sales Algorithm Pseudocode*

Text

Description automatically generated

**Web-Scraping Data Collection**

Even with all the missing data interpolated, comparisons between the financial data of different films are still of little worth. This is because of inflation. A dollar in 1962 was worth over 9 times the amount of a dollar in 2021. It’s clear that for a realistic analysis, inflation must be accounted for. The easiest way to accomplish this task is with the consumer price index. This is a value, known as cpi, is computed monthly and averaged yearly by the US Bureau of Labor Statistics. It measures the average change overtime in the prices paid by consumers for goods and services. Its extremely useful because it can be used to calculate the value of a dollar from one year, in any other year. Specifically, the formula is the base price in the original year multiplied by the ratio of the new year's cpi divided by the original year's cpi: new amount = original amount \* (new cpi/original cpi). The application of this equation allows the program to easily convert the worldwide gross that is contained in the international revenue column into 2021 dollars.

Just because cpi is a useful statistic does not mean it is available on IMDb's API. In this case, the data was found on the Federal Reserve Bank of Minneapolis web page. The URL is as follows: https://www.minneapolisfed.org/about-us/monetary-policy/inflation-calculator/consumer-price-index-1913-. A quick travel to that site will reveal a very simple table containing only the year, the annual average cpi for that year, and the rate of inflation for the same. The python library beautiful soup was used to return the html code of the document which was then scraped to produce a dictionary containing the year and it's corresponding cpi. The code that does this is found in the 11th block of the project007.ipynb file. The pseudocode for the segment can be found below.

*Web Scraping Pseudocode*

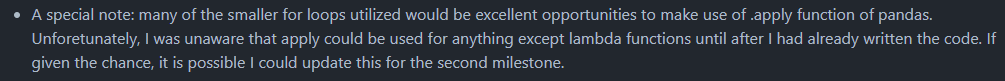
Text

Description automatically generated

**Calculated Data**

Even with all the data derived from web-scraping and API calls, there was still some values that were obtained by other means. In this final instance, data was calculated. Specifically, the 'Adjusted Revenue' and 'Average Rating' columns. The average rating is exactly what it sounds like; it is the sum of the IMDb user rating and the Metacritic score divided by 2. However, both sets of data had to be converted from strings. In addition, the user rating was multiplied by a factor of 10 to put both ratings on a uniform scale of 100. More interesting is the adjusted revenue. This value is derived from applying the formula described in the web-scraping section, (new amount = original amount \* (new cpi/original cpi)), to the international revenue to produce an inflation adjusted total revenue for each movie. With those final two calculated data columns, the project007 database can be considered complete.

*A Special Note*



***Column Documentation***

**Title**

The data stored in the title column is the name of the film.

Type: *String*

Source: *API Call*

**Year**

The data stored in the year column is the year in which the film released.

Type: *String*

Source: *API Call*

**Actor**

The data stored in the actor column is the name of the person who portrayed the main character, James Bond, in a particular movie. It represents the leading role of the film.

Type: *String*

Source: *API Call*

**US Box Office**

The data stored in the US Box Office column is the amount, in United States Dollars (USD), that film grossed in its theatrical run in America. It represents the value of all the tickets sold in the United States.

Type: *String*

\*floater(input\_str) converts to float data type

Source: *API Call*

**International Revenue**

The data stored in the international column is the sum, in USD of the foreign and domestic box markets for an individual movie. Essentially, it is the worldwide gross revenue of the film. It represents the total value of all theatrical tickets sold on Earth.

Type: *String*

\*floater(input\_str) converts to float data type

Source: *API Call / Interpolated*

**Adjusted Revenue**

The data stored in the Adjusted Revenue column is the international revenue data, converted into 2021 USD. The coefficient of inflation is determined with the web-scraped CPI data and applied to the total worldwide revenue data to produce the correct, inflation-adjusted values. This represents the amount of money the film would have made were it released this year.

Type: *String*

\*floater(input\_str) converts to float data type

Source: *Calculated / Web-Scraped*

**Average Rating**

The data stored in the Average Rating column is sum of a film's IMDb Rating and Metacritic values divided by two. It reprsents the mean of the Metacritic score and IMDb user rating for each movie.

Type: *String*

Source: *Calculated*

**IMDb Rating**

The data stored in the IMDb Rating column is each film's average user score on a 10-point scale. It represents the mean value of all IMDb member's review scores for the film in question. It is displayed in ## / 10 format, with 10 being the highest and 1 being the lowest.

Type: *String*

\*replace\_rating(input\_str) converts to percentage and returns a float data type

Source: *API Call / 1 Interpolated data element*

**Metacritic**

The data stored in the Metacritic column is the mean of the aggregate score of many published critical reviews. It is returned as a percentage ranging from 0% to 100% and represents the average score of all professional reviewer's scores.

Type: *String*

\* replace\_critic converts to float data type

Source: *API Call*

***Questions***

**Empirical Questions:**

1. What 007 movie(s) grossed the most money, both in actuality and when inflation is accounted for?
2. Which leading actor's films made the most money, both the nominal amount and when inflation is accounted for?
3. What movie was the best reviewed: by critics, by IMDb users and overall?
4. Which leading actor's films were the best reviewed: by critics, by IMDb users and overall?
5. Which leading actor portrayed the character in the most films?
6. Which leading actor portrayed the character for the longest period of time?
7. What is the average time in between each James Bond films debut?
8. Which decade saw the most releases of 007 films?
9. Which leading actor had the shortest time in between each of his movies?
10. What is the average rating for all 007 movies: by critics, by IMDb users and overall?
11. What is the average discrepancy between critic scores and user ratings?
12. Which films has the highest and lowest discrepancy between the two ratings?

**Analytical Questions**

1. Do better critical reviews correlate to a higher revenue?
2. Are higher user ratings associated with increased ticket sales?
3. Is the user rating or critics score more important in regard to revenue?
4. Does the data support the theory that the decision to recast the role is, at least in part, financial?
5. Is it likely that review scores are a factor in deciding to replace the leading actor?