CMPT 353
PROJECT SUMMARY

Insights Gained from WikiData and Movies Dataset

Jonathan Sawali
301262582

Introduction

This report focuses on how large amounts of movie data can be used to gain specific insights and potential business strategies that may be useful for the media industry. The movie data used was collected from **WikiData**, **Rotten Tomatoes**, and **IMDB** datasets provided in the course website. We will cover the extent of work and effort that was needed to extract, clean, and analyze the data to gather valuable insights. The main question to be answered is the relation between cast members and directors to a movie's profit. Other factors like publication date and genre might also be a factor to a movie's profit.

Cleaning the Data

The WikiData file in `wikidata-movies` given to us were messy and not human-readable. The list of entries in columns such as `genres`, `cast_members`, or `directors` were in WikiData ID, that had a format of the letter `Q` followed by a sequence of integers (e.g. `Q8923722`). The WikiData IDs have mappings to its actual string entries in the collection of gzipped JSON file called `label_map.`

The PySpark program `build_useful_movies.py` was written to load both wikidata-movies and label_map into the program, and map all relevant columns from wikidata-movies into its respective labels as written in `label_map`. The operation took quite a bit of time considering that the list of entries in each row had to be exploded using the `explode()` function, joined, and grouped again by its movie name. The program also output a new smaller dataset called `movies-readable.json.gz` that has all the relevant columns replaced with actual strings of names instead of WikiData IDs. The output was coalesced into one compressed JSON file for ease of use in other parts of the analysis that uses Pandas/NumPy instead of PySpark.

Analyzing the Data

`correlations.py`

The program correlations.py was written to find correlations between cast_member, critic_average, audience_average, and made_profit. The correlations were done in pairs to see if critics' and audiences' ratings had any effect on the movie profit. Below are the pairs and correlation coefficients of each pair, resulted from `.corr()` function in Pandas.

Finding the correlations gives a little insight on how the data connects with each other. Although there are some columns that had surprisingly low correlation coefficients, such as the first two pairs of `cast_member` and `director` with its relation to `made_profit`. Instinctively, `cast_member` or even director should probably have some more correlation to the movie's profit status, but the data says otherwise. Potential issues that may cause this vary from the number of casts having to much variation, to the data not having a lot of `made_profit` values to compare.

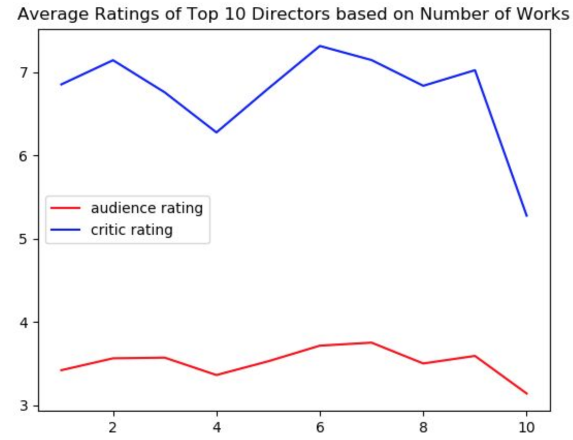
The pairs of audience_average and critic_average have a relatively good correlation to a movie's profits. The data suggest about 20% correlation for `audience_average` and 13-15% for `critic_average`. One might take the conclusion of the audience having more control than critics to how profitable a movie can be. The more the audience likes the movie, the more willing they are to spend on a movie ticket. As for 20% being 'good', it is important to take into account that people have very wide preferences on movies and how they perceive media. A 20% correlation in the data is relatively good in this case.

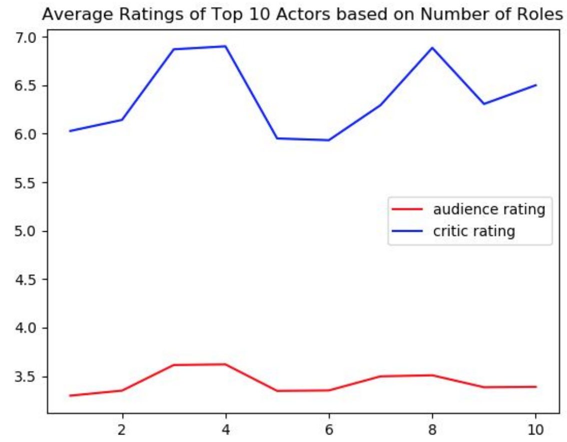
Column 1	Column 2	Correlation Coeff.
`cast_member`	`made_profit`	0.00290
`director`	`made_profit`	-0.0065
`audience_average`	`made_profit`	0.210
`critic_average`	`made_profit`	0.155
`audience_percentage`	`made_profit`	0.210
`critic_percentage`	`made_profit`	0.133
`audience_average`	`critic_average`	0.699
`audience_percentage`	`critic_percentage`	0.687

analyze.py

Now that correlation on data is established, it makes sense to start looking for more significance in the data. This Pandas/NumPy program takes in two compressed JSON files which are `movies-readable.json.gz` and `rotten-tomatoes.json.gz`. These two data frames were joined, and a subset of columns were selected for analysis. Key columns in this analysis were `cast_member`, `director`, `audience_average`, and `critic_average`. The purpose was to group the data by `cast_member` and `director`, count the number of movies they have directed (or have been in), and display the average rating and standard deviation for each group.

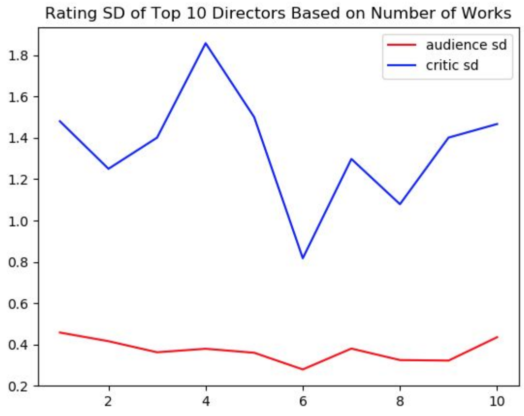
The results strongly suggest that critics give higher ratings to movies on average than the audience. The analysis was done two different times, one with directors and the other one with `cast_member`. The result is sorted by how many movies a director had directed, and how many movies an actor/actress had performed in. Below is a side-by-side plot on the top 10 entries from the result of both analysis (`cast_member` and `director`).

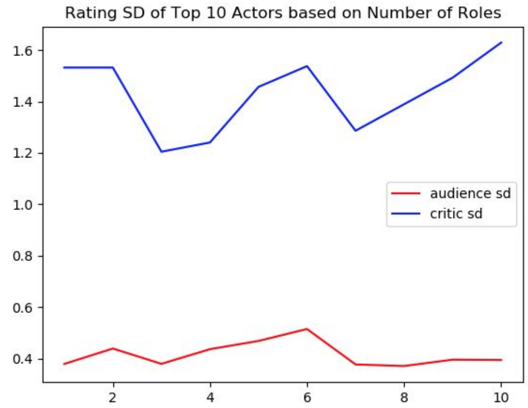




As you can see in the two diagrams above, critic ratings seem to have a significantly higher average than audience ratings. This may be caused by a number of factors. One possible factor is that audiences seem to give ratings without subjective thought. One person might give a movie a 0 if he/she did not like it, but may also give a 10 for a movie they like. The ratings can have a huge skew and may not represent the actual movie quality. Critic ratings are usually better because critics really take into account the objective qualities of each movie.

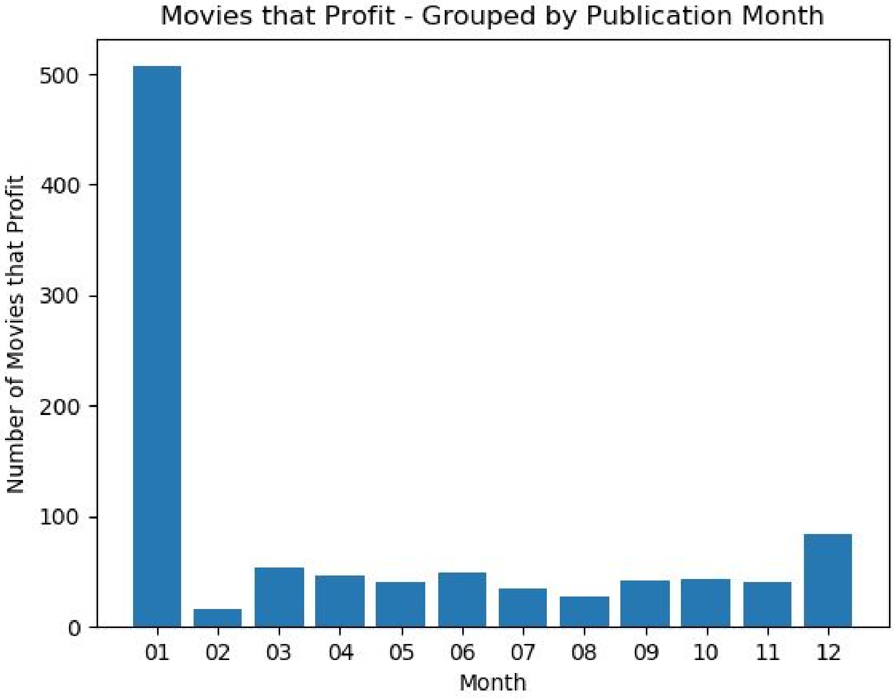
However, an interesting fact comes up when we put the standard deviation of each group into account. The graphs below show the standard deviation of each group. The audience ratings have a smaller number of standard deviation compared to critics.





profitable_time.py

This Pandas/NumPy program loads the same data as the other two programs, but approaches it in a different way. analyze.py tried to gather profit insight from cast members and directors, but more insight may be gathered from other parts of the data like publication_date. Each entry in the WikiData dataset comes with a publication_date column. profitable_time.py groups them by month of publication and aggregates them by counting the number of movies that made profit during said month. The results are presented in the histogram below.



The results were very surprising. According to the given data, movies/works that made profit is published in January. The results clearly had a certain bias towards the first month of the year. Not too much confidence can be put in the data since only a small part of the dataset had information on profit. A different result might emerge if the dataset had more information on profit.

Conclusion

Gathering insight and getting results from a dataset can present more challenges than expected. The initial state of the data were quite disorganized and borderline unusable. After a number of cleaning steps and organization, the data is finally ready for analysis. The analysis was done in several steps. The first step was to establish a correlation in the data, the second step was to dig deeper into the data, and the last step was to find extra detail in the data we find.

On a very high level, the analysis found several key insights on the movie data:

- Critic average ratings and audience average ratings were significantly correlated with each other
- Movie ratings have a relatively good correlation to a movie's profits in the context of our data
- Critic ratings can be assumed to be more objective compared to audience ratings, making critic ratings a better gauge of how a movie is received
- January is the most profitable month to publish a movie (in the context of our data)

Limitations and Problems

In the process of execution, the project encountered a number of problems. The problems vary from the data itself to the limited amount of time that was given. Initially, the project was about predicting ratings by using genres, cast members, and directors. The idea was scrapped due to the genres being poorly classified and not very workable. Some movies of the same genre would have different strings that represent them (e.g. Two drama movies having the genres 'drama film' and 'drama teenager'). Working with genres would require a lot of manual data cleaning that would take up a large amount of time.

Predicting the ratings of a movie also had its own challenges. Naive Bayesian classifiers were used to try and predict the scores using cast member and director data, but had less than acceptable results. The scores were not satisfiable, especially with that much data involved.

If the project had more time, it is definitely possible to get more interesting results. A possible question to be answered is if movie genre popularities changed over time, or if certain actors/actresses are type-casted into certain genres.

Accomplishment Statements

- Extracted, cleaned, and re-organized large datasets into smaller and usable chunks of data, lowering data load times significantly
- Designed and implemented a work plan to gather financial details in the data, potentially usable by movie marketers and analysts
- Visualized and summarized findings, increasing readability and scope of audience