# Movie Review Metrics vs Box Office Gross Values

DNSC 6211: Programming for Analytics

Abhinav Chandel
Tingting Ju
Xinyi Wang
Yunning Zhu
Daniel Chen

### Abstract

Describe your project within 200 words. One way is answer the following questions regarding your project: (a) what did you do? (b) why did you choose do that? (c) how did you go about doing your project (d) what did you find out?, and finally (e) What did you find out? The content of your abstract and the outline and contents of your report may vary according to the needs of your specific research topic.

# Contents

# 1 Introduction

For full-time graduate students, going to the movies is perhaps the most available and low-cost pastime option. We often check movie ratings on popular review sites such as Rotten Tomato, IMDB, Metacritic, etc. to help us to decide whether or not we should go to see a particular movie. Therefore, it is natural to expect that higher-rated movies are likely to generate more moviegoers, and consequently, more box office successes. But is that truly the case?

We decided to conduct our project by examining the relationship between movie review metrics and box office gross values. More specifically, we decided to look at all the movie reviews from 2015 and see if the top-rated ones actually ended up being the top 100 grossing movies of 2015. The problem we identified is: How much predictive power does movie review metrics have over box office success (in terms of both worldwide and domestic revenues)?

# 2 Background

The expected storyline would be something like: Rotten Tomato Tomatometer ratings are good predictors of box office gross values. or Metacritic ratings are not good predictors of box office gross values. Or The combination of Rotten Tomato Tomatometer and Metacritic ratings has the best predictive power of box office gross values. The underlying data sources that drive our storyline would be as follows:

1. Revenue-Related Data:
   (a) Box Office Mojo
2. Budget-Related Data:
   (a) The Numbers  Movie Budget
3. Review-Related Data:
   (a) Rotten Tomato: ratings by audience
   (b) IMDB: ratings by audience
   (c) Metacritic: ratings by critics
   (d) Twitter: commentaries from general populace

Most of the data could be obtained via web-scrapping. Data from Twitter and/or Facebook would require the use of API. And they would have to be further cleaned for quality metrics that include feelings and satisfaction ratings, which would then be translated into sentiment scores. After cleaning up all the datasets, we would like to load all of them into a central database via SQL.

# 3 Method

Our proposed methodology is as follows:

1. Data Gathering: Scrap revenue and review data off web in Python
2. Data Transformation: Create sentiment scores in Python; normalize and standardize rating scores in Python
3. Data Cleansing: Ensure clean DataFrames that can easily be output into csv files
4. Data Consolidation: Load all datasets into one database via MySQL
5. Analysis: Build simple linear regressions / multiple regressions in R

6. Presentation: Display results via visualization (ggplot and matplotlib) and interactive web applications (shiny)

There are some foreseeable concerns:

- Sentiment Analysis: Facebook was initially identified as one of our data sources. However, we may have to abandon this data source due to our inexperience with Facebooks API.
- Time Constraint: Our project scope is ambitious with numerous data sources. We fear that we will not have ample time to complete our project.
- Regression Results: What if none of our regression models turns out to be statistically significant?

# 4 Organization

Our division of labor is fluid  we will shift the workloads around the team to fit our individual schedules as we go through the project. Currently, our division of labor is as follows:

- Abhinav: Project idea formulation, web scrapping (ratings), sentiment scores
- Yunning: Project idea formulation, web scrapping (ratings), sentiment scores
- Xinyi: Project idea formulation, web scrapping (ratings), sentiment scores
- TingTing: Project idea formulation, web scrapping (misc.), initial data consolidation and regression
- Daniel: Project idea formulation, web scrapping (revenues), report/presentation slide write-ups

## 4.1 Workflow

Provide a diagram of the workflow for your project. The command to include a diagram is shown below. Make sure you remove the comment and change the name of the graphic file without extension. Also, instead of **Quick Build** choose **PDFLaTex** from the dropdown option. Then generate the pdf with **View PDF**.

Please explain your workflow diagram in this space. Limit this to 250 words.

## 4.2 Project structure

Describe your data sources. In addition, describe how they are related to each other and to the research question(s). Limit this to 250 words.

## 4.3 Figures and Tables

List your tables and figures and explain why you chose to use them. Explain how these tables and / or figures contribute to your "story." Limit this to 250 words.

# 5 Discussion

This section requires you to discuss your experience. Describe the value of your project. What are two main "selling points" of your project. Limit this to 150 words.

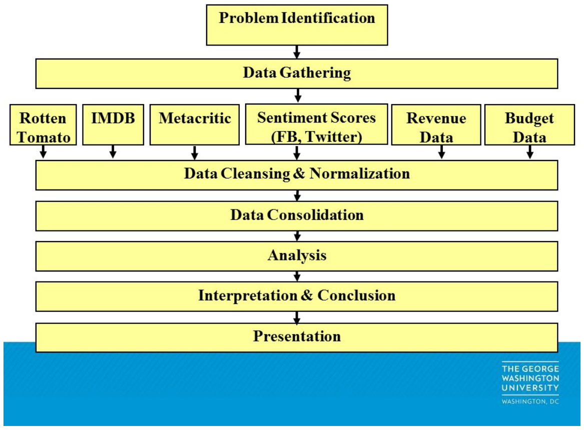

Figure 1: Our projected workflow

### 5.1 Learnings

Discuss some of your ”better moments” in this projects - the ones you enjoyed. Also describe. what you learned in this project. Limit to 150 words

### 5.2 Challenges

Discuss some of your ”difficult moments” in completing this project. You may want to write about things you wanted to do but could not complete and why. Limit to 150 words.

## 6 Bullets and numbered lists (FYI, delete in your report)

This is up to you. If you want to add another section. This section explains how to make lists. In you final report you should delete this part.

### 6.1 Bulleted and Numbered Lists

LaTeX is very good at providing clean lists. Examples are shown below.

- Bulleted items come out properly indented and spaced, every time.
  - Sub-bullets are a virtual no-brainer: just nest another `itemize` block.
  - Note how the bullet character automatically changes too.
- Just keep on adding `\items`. . .
- . . . until you’re done.

Numbered lists are almost identical, except that you specify `enumerate` instead of `itemize`. List items are specified in exactly the same way (thus making it easy to change list types).

1. A list item
2. Another list item

3. A list item with multiple nested lists
    - Nested lists can be of mixed types.
    - That's a lot of power and flexibility for the price of learning a handful of directives.
        (a) Like nested bullet lists, nested numbered lists also "intelligently" change their numbering schemes.
        (b) Meanwhile, all *you* have to write is `\item`. LaTeX does the rest.
4. Back to your regularly scheduled list item

BTW, this is a great site to generate tables in Latex and learn how to do it in Latex – `http://www.tablesgenerator.com/`

# 7 Conclusion

Wrap up your paper with an executive summary of the paper itself, reiterating its subject and its major points. Limit this to 150 words.