**Project Title**

Movie Rating and Review Data

**Team Members**

Nerojan, Prateek, Jialong, Edward

**Project Description/Outline**

We are looking to aggerate moving ratings and reviews for films in 2019 form IMDB, Rotten Tomato and NYT move reviews into one data base in SQL.

**Data Sources**

<https://www.imdb.com/> (Web Scrapping)

<https://rapidapi.com/rapidapi/api/movie-database-imdb-alternative/details> (API)

<https://pypi.org/project/rotten_tomatoes_client/> (API - if we have the time)

**Rough Breakdown of Tasks**

Nerojan – API, Data Cleaning, Loading

Prateek - Web scrapping, Data Cleaning, Loading

Jialong - Web scrapping, Data Cleaning, Loading

Edward – API, Data Cleaning, Loading

**Process**

**Extract**

1. Used beautiful soup to scrap movie names and IMDB ratings for all movies in 2019
2. Then used the movie names obtained above from IMDB to get the New York Times movie reviews, critics picks, critics names, headline and movie summary using an API. Movie names from IMDB remains the key
3. Then we used the same names obtained from IMDB to feed the Rotten Tomato package. Package is open source designed to get around the bureaucratic API key obtaining process. From Rotten Tomato we obtained movie ratings (i.e. meter class and meter score)

**Transform**

1. We collected the data in lists and JSONs using IF statements to collect only complete data sets (i.e only collecting data where all columns are available from the API and/or web scrape)
   1. We focused our efforts on the data obtainment process where we reduce the need to clean the data by only obtaining complete and clean data
   2. Downside to this is we lose data points but incomplete data points are not usable nonetheless
2. Then we joined the lists into a pandas Data frame
3. Dropped dupes where necessary using pandas

**Load**

1. Created schemas in SQL for the three data frames we created in pandas
2. Used sqlalchemy to upload the data to SQL
3. Used JOINS to gather the data from all three different sources and display them in one table