Extraction, Transformation, and Load Technical Report

**SENTIMENT ANALYSIS OF JOKER MOVIE REVIEWS**

Hulester Bing

Antony Gikanga  
Subodh Khanal

Robert LaBelle

Aman Garg

Changler Dunklin

|  |  |
| --- | --- |
| **1.** | **INTRODUCTION** |

# 1.1 Summary

The objective of our ETL project was to scrape rottentomatoes.com for user reviews of the movie *Joker* so that we could upload them into a mongo database and conduct sentiment analysis to better understand how the movie was critically received by the public.

# 1.2 Scope

Our data was sourced from the text of over 44,000 user reviews of the *Joker* movie located at rottentomatoes.com.

# 1.3 Technologies and resource contributions

*This section lists out the team members and their contributions towards the ETL initiative. Use this section to also outline (or list) the tech stack used to obtain the final outcome.*

* **Selenium** and **ChromeDriver** were utilized to scrape multiple webpages of user reviews from rottentomatoes.com*.* Once we gathered the html responses, we used **Beautiful Soup** to parse through the data and gather the contents of each user review. Then we used **TextBlob** to analyze the subjectivity and polarity of each review and assign a set of corresponding scores. Lastly, we loaded the user reviews, their polarity scores, and their subjectivity scores into our Mongo database by using **PyMongo**.
  + Hu and Antony focused on scraping the data from Rotten Tomatoes
  + Amman and Sabode were responsible for parsing the HTML response with Beautiful Soup and writing the code for sentiment analysis.
  + Chandler and Robert were responsible for loading the data in MongoDB.

\*While we had individual responsibilities, all team members helped each other on all parts of the project.

|  |  |
| --- | --- |
| **2.** | **ETL DETAILS** |

*This section outlines a more detailed description of the processes utilized/proposed to achieve the objectives of this initiative.*

# 2.1 Data Import/Extract Sources and Method

We used Selinium and ChromeDriver to extract the html responses from multiple webpages so that we could gather a large sample of user reviews. After collecting the contents of each review, we assigned an index value to each record, along with the corresponding polarity and subjectivity scores. Then we used PyMongo to import these records into our Mongo database.

# 2.2 Data Acquisition

A large sample of user reviews was required to understand the critical response of RottenTomatoes’ users to the movie. The Joker is expected to leave theatres in 2 weeks, after which we would not expect many additional reviews to be submitted. Because the data on rottentomatoes.com can be dynamically updated with user reviews, a strategy for updating our dataset would require the verification of new record entries by checking them with our previously recorded dataset. When updating our dataset, we would first verify that the contents of the new user reviews are not identical to the contents that have been recorded in our Mongo database.

If someone wanted to use our code to analyze sentiment analysis on a different movie on Rotten Tomatoes website they could substitute the URL and our code would work on any given movie that they selected.

# 2.3 Data Transform

As stated above Beautiful Soup and TextBlob were our primary tools for transforming the data so we could record the contents of each review along with the corresponding scores related to polarity and subjectivity. Beautiful Soup was utilized to parse through the contents of our html response so that we could gather the contents of each user review. TextBlob was then utilized to asses the polarity and subjectivity of each review and assign a set of corresponding scores.

# 2.4 Data Integrity

Our data was sourced from RottenTomatoes’ user reviews. It is possible to falsify a review or for someone to provide a review who hasn’t seen a movie. However, given that we gathered over 44,000 reviews we’re confident that the dataset is large enough to outweigh any false records. Although new user reviews may be continuously uploaded, it is not necessary to update the local data unless the most current state of user reviews is required for analysis.

# 2.5 Data Refresh Frequency

Again, given that the Joker is leaving theatres soon, we expect the number of new reviews added to fall precipitously. Updates to the dataset would not be necessary unless the most current status of user reviews would be required for analysis. In that case, the database could be updated by verifying the contents of new record entries as specified above.

# 2.6 Data Security

Our data only scraped the contents of reviews located on a public forum and no PII (personally identifiable information) was acquired.

# 2.7 Data Loading and Availability

All of the recorded data was exported to our Mongo database. This database would be accessible to users. We later reloaded the MongoDB database into a Pandas data frame for further manipulation, visualization, and analysis.

|  |  |
| --- | --- |
| **3.** | **DATA QUALITY** |

Address in this section success criteria for this project. Summarize the parameter KPIs such as Totals and expected counts. What user acceptance testing was performed and what were the outcomes. What is the recommended site acceptance testing that your client can perform to ensure the expected outcomes meets their expectations?