**MoodFlix**

**Team name**: Team Jay

**Members**: Asheton Harrell, Kelsie Belan, Phillip Griggs, John Little

It’s been said before that it’s not what you say that people remember, but how you say it. With online reviews we remove the part of how you say it, which makes the tone subjective to the reader of how they perceive a critics review.MoodFlix aims to add another human aspect to movie reviews. We want to add the emotion to the review by using multiple APIs and sentiment analysis. If successful we would create more depth and a stronger connection between the review and reader.

To add emotion to these reviews, we first needed to know the emotion of the review. Using IBMs Watson API, we were able to use the tone analyzer function to run sentiment analysis on the text of the reviews. We provided the reviews to Watson using the New York Times API for movie reviews. We were originally going to use Twitter’s API to scrape tweets about movies but ran into multiple issues that made it unrealistic to continue. To get the New York Times reviews we made a cURL request for a particular movie and then stored the results into an array. Then we made a cURL request and passed our reviews to the Watson API which ran sentiments analysis. Watson then returned a JSON format reply which we then parsed and graphed using JavaScript. We used a database to act as a cache to store previous searches of users.

In the early stages of this project, we thought using Amazons Mechanical Turk service would be a good idea but quickly realized consistency would be an issue. Since these reviews are subjective to the reader, we couldn’t guarantee consistent analysis of emotions. That is why we settled with using Watson. While Watson may not be completely accurate, it is at least consistent. Another issue mentioned was using Twitters API or YouTubes API to scrape tweets and comments about the movie. Using those APIs to search a movie called “Cars” would return various results for more things than just the movie “Cars.” The search returned anything that talked about cars. Even if we were able to get only reviews or comments about the movie, we then would have to filter out sarcastic comments or trolls. Using the New York Times API, we still were limited by what movies they had in their database and their reviews. Our overall goal was to use multiple APIs to gather reviews and use sentiment analysis to get an average sentiment of a movie. Due to multiple limitations that were discussed above, this project is nothing more than a proof of concept.

Tl;DR

|  |
| --- |
| **What is new/interesting?** Sentiment analysis on movie reviews to add more emotion and context to reviews. No other website offers this feature, that we know of.  **A database is required:** MySQL, used to act as a cache.  **A JavaScript and CSS-based front-end is required:** Used JavaScript to graph and CSS for homepage styling.  **A PHP-based back-end is used**.  **Real-world data is required:** Used the New York Times API for movie summary and reviews.  **At least 1 external web service or API needs to be used:** New York Times and Watson by IBM. |
|  |