National and Kapodistrian University of Athens

Department of Informatics and Telecommunications

Web systems and applications - Project progress report
Summer semester 2023-2024

Student 1	Triantafyllou Thanasis
	7115132300010
Student 2	Tsiompikas Dimitris
	7115112300036
Student 3	Syrios Konstantinos-Zois
	7115112300035

Movie Reviews Sentiment Mining and Analysis

Web Crawler implementation

For this project we wanted to crawl movies in IMDb and scrape the reviews(or a subset) of each movie. The first problem we encountered was that the ip got blocked when we were crawling and scraping the reviews, since the robots.txt of IMDb disallows crawling/scraping, for that reason they provide an API and a python library that gives access to their data but that was out of the scope of our project. The second problem was with the pagination. In the main review page we have 25 reviews and every time a user presses load more in the reviews section asynchronously more html gets loaded (AJAX) to make visible 25 more reviews. So to counter both those problems we decided to use selenium and approximate how a user would use IMDb. This idea was successful with the downside that the crawler with the bot is slower. We decided to target 1.000.000 movies to crawl and take 100 reviews from each. If a movie has less than 100 reviews we take all of them. The next step is to make a second bot/crawler that will gather data from movie 1.000.000 and move downwards so 2 crawlers will gather data from iMDB. We also search for a second movie review site that will not block our ip , and try to crawl without selenium to create a third crawler that will also work in parallel with the other 2. Our first test with only 1 bot/crawler in IMDb provided us with a dataset of around 20k reviews in 6 hours of crawling. We hope to double our dataset in the same crawling time.

Dataset Management

After scraping the data from website/s (presently just from IMDb) and storing them as json files we process them in order to create two datasets to use in training the machine learning models.

For the basic dataset we labeled the text reviews using (negative, neutral, positive) labels based on the review's corresponding numerical rating. IMDb uses a 0-10 rating system and we divided the sections as follows:

- [0, 4] as Negative
- (4, 6] as Neutral
- (6,10] as Positive

The above is saved as a separate json file to be used independently from the rest of the program.

In addition, we experimented with the idea of detecting positive/negative keywords in the review text and then using their counts as a secondary feature for the models.
We start by splitting the negative and positive keywords from this kaggle dataset. Then we iterate through all the reviews detecting which keywords are used in each one. For each "hit" we associated the keyword with the review's ID, to create a kind of index, with the keywords as the keys. After that we determine the 100 most common negative and positive keywords in the reviews and use their counts as the secondary features.
The intermediate lists (e.g. positive keywords-reviewIDs) as well as the "extended" dataset are saved as separate json files to be used independently.

Machine Learning Models

For the next part, we will implement 3 machine learning models and compare their performance on Sentiment Analysis for our dataset. The models will be Logistic Regression, SVM and Naive Bayes. We will use the scikit-learn library to implement them and also create some visualizations for their performance and the dataset itself. We will use several techniques for pre-processing the text data from the reviews such as removing stop words, special characters and numbers, lemmatization and lower-casing the text data in order to get the optimal performance and also find the best hyperparameters for our models.