CS410 Project Proposal
RMK Group (Rui Mao, Matt Malitz, Karl Vosatka)
Team Leader: Karl Vosatka
Prof. Cheng Zhai
10/23/2022

Our group is RMK, and consists of Rui Mao (ruim6), Matt Malitz (mmalit4), and Karl Vosatka (vosatka2). We are going to build a sentiment analysis model for the social media film review website Letterboxd that will rate a movie as “above average” or “below average” based on a sentiment analysis of some of its most popular reviews. This is interesting to us because a lot of numerical rating systems for users can provide murky information masquerading as data. At best, numerical rating systems are providing an ill-defined, entirely subjective categorization masquerading as data. There is no standard definition for a “2-star” movie, nor for a “5-star” one. Also, users may bias towards extreme positive or negative ratings across films they watch. Applying sentiment analysis to popular reviews allows us to gauge the merit of a film based on the richness of text data. This might allow users of our model to get an alternative sense of whether a movie is really worth watching or not.

For our project, we will build a Java applet with a sentiment analysis model and web scraper in the back end. Upon user entry of a movie title, the web scraper will locate the appropriate movie page and its written reviews and collect a large sample of them for analysis by the sentiment model. The sentiment model will then rate the sentiment of each review and return some results from the analysis. At minimum, the model should return to the user a rating of the sentiment analysis across reviews as either “above average” or “below average”(i.e. good or bad). These results will be displayed in the front end, possibly with more details about various reviews, hyperlinks, the average movie rating, etc. As such, we will use Python in the backend and Java in the frontend.

To build a scraper, we intend to employ tools like the python libraries BeautifulSoup and Requests to take in a movie title, find the appropriate link on Letterboxd, and scrape the first 500 or so written reviews, along with any ratings provided with the review. We will then collect the reviews in a PostgreSQL database for reference. We anticipate that this aspect of the project will take about 20 hours or so, though it is possible it will take less. Karl will be principally responsible for this component.

Then, we intend to construct a sentiment analysis model with reference to existing Letterboxd data hosted on kaggle.https://www.kaggle.com/datasets/samlearner/letterboxd-movie-ratings-data?select=ratings_export.csv. We plan to use Python and the NLTK package to achieve this. We expect this aspect of the project will take 20 hours or more, and will be conducted as a collaboration between Rui and Karl, with Rui taking charge.

Once we have established a model for sentiment analysis and a scraper for analyzing each review from a URL, we will map that to a Java applet front end interface via the JavaFX library. The front end will include example articles to run sentiment analysis on, a search bar to peruse articles we can perform sentiment analysis on, and navigation tools. The way the frontend will interact with the sentiment analysis program written in python is by using an abstraction of Fork Exec called ProcessBuilder. This will allow us to launch the sentiment

analysis program with the specified article by passing the article as a command line argument Once this is done the output of the sentiment analysis will be stored in a file which will get its information outputted to the GUI. We may consider augmenting the capabilities of the applet to include a stored database of previous sentiment analysis data to represent upon search. The program will be compiled using Maven. This component of the project should take 20 hours or more, and will be Matt's main responsibility, with collaboration for the process of making python code compatible with Java applet tools.
Here is our list of tasks as stated above with time estimates:

- Build a scraper (20 hours, Karl)
    - Construct a crawler for website reviews on a per-movie basis using BeautifulSoup and Requests (At least 10 hours)
    - Collect review data in a PostgreSQL database (At least 10 hours)
- Build Sentiment Analysis Model (20 hours, Rui)
    - Investigate the sentiment analysis and build the module structure (At least 10 hours)
    - Model implementation (At least 10 hours )

- Wrap the above in a Java Applet (20 hours, Matt)
    - Build the frontend scenes that will be used in presenting the sentiment analysis data and URLs of the articles.
    - Design the backend to launch the sentiment analysis program that is written in python. This task should also handle all race conditions between threads.
    - Potentially add other features such as a search bar if time permits.