1.
Captain - Roshan Mahesh (roshanm2)
Jimmie Gann (jcgann2)
Keyuan Chang (keyuanc2)
Vedang Bhargava (vedangb2)
Vaasay Farid (vfarid2)

2.
Topic and Approach

We are going to do a sentiment analysis of movie reviews. The task is to look past the review rating and focus on the overarching theme (positive or negative) of the review. This is important for users who are deciding whether or not to watch a movie since the movie review rating that people give usually isn't representative of their written thoughts of the movie. Additionally, most of the time people just want to know whether a movie is good or not and don't want to read extremely long reviews. Providing a sentiment analysis on movie reviews allows users to understand the population's view on the movie without spending the time to read each review.

Our planned approach to creating the sentiment analysis is to create a model that analyzes movie reviews using a training set that trains the model. We can then feed in movie reviews, and the output would be whether or not the movie has more positive or negative reviews. We can create a model using the Naive Bayes/bag of words approach as well as analyzing the movie reviews through a unigram or bigram model.

Datasets

IMDB Dataset
Stanford Databset

The former is a dataset provided by IMDB and is the raw data from 50,000 movie reviews across their website. The latter is also raw movie reviews from Stanford that have been preprocessed to determine if a movie has a positive or negative sentiment, this will be used as our training data.

The outcome of all of this is to accurately tell whether the reviews for a movie are more positive or negative. We hope to attain an accuracy level of at least 80%. We are going to evaluate our work based on our accuracy level to the actual sentiment of the movie reviews, which is why we want to attain an accuracy level of at least 80%.

3.
Language and Libraries

We plan to use Python for this analysis because of the extensive libraries and tools designed for NLP and machine learning. For example, we could easily make use of the following libraries: pandas (for data preprocessing), numpy (efficient way to manipulate data), metapy (for text analysis), pytorch (for training and testing the model), seaborn and matplotlib (for data visualization for our findings ie the model performance), etc.

Performance Metrics

Quantitative Evaluation: Accuracy, Precision, Recall, F1-score
Qualitative Evaluation: Analyzing false positives/negatives, and understanding which kinds of reviews the model struggles with.

4.
Detailed Tasks and Workload

Given that there are 5 students in our team, we plan to spend

- 5 hours of reading about this topic to understand sentiment analysis
- 10 hours studying and reviewing our datasets to fully understand what we are working with
- 5 hours of data collection
- 10 hours with data processing
- 5 hours for model comparison and selection
- 20 hours coding the Naive Bayes algorithm for the sentiment analysis
- 10 hours with model training
- 5 hours analyzing the results
- 5 hours for peer reviews and feedback analysis
- 10 hours revising the project details and making any corrections
- 15 hours with the presentation component

During the course of the project, we know that something may arise that ends up taking much longer than we originally anticipated. The number of hours that we're anticipating is fully dependent on the level of accuracy that we can get our sentiment analysis to. If we're only at around 50% accuracy, then we're going to have to spend a lot more time revising our Naive Bayes algorithm.