Project Proposal: Sentiment Analysis for Yelp reviews to predict rating given the review and prevent pollution of reviews.

Team Members:

- Patrick Han
- Daniel Tang
- Michael Price

Problem Statement: Often times Yelp reviews can be misleading. Passionate users will often give a strong review but leave a star rating that does not necessarily reflect their words. Many users of the Yelp app probably do not take the time to read reviews, and instead browse restaurants based purely on star ratings. Unfortunately, these misleading reviews have the potential to pollute the restaurant's reputation. We seek to develop a tool for Yelp moderators to flag potentially misleading reviews.

Previous Work: Sentiment analysis is very important for platforms like Twitter and Facebook to counter misinformation.There should be a decent amount of literature regarding this topic, however, almost none of the sites employ an algorithm that may warn a user if their post contains misleading content.

Dataset: The dataset was provided by Yelp, and can be accessed through this link: https://www.yelp.com/dataset/download. The download includes these JSON files:

- business.json
  - Contains information about the business being reviewed, including names, location, and average star rating.
- review.json
  - Includes information about the review itself, including the star rating, user id, and the text of the review
- user.json
  - Contains information about the user who wrote their review, including their name, average star rating, and number of reviews they have written.
- tip.json
  - Tips are short tidbits about that reviewers post about a business. Contains the user information and the text from the tip.

Preprocessing:

- The Yelp dataset will be downloaded and the json file containing the reviews will be trimmed for review (String) mapped to its corresponding rating (float).
- We will also cut down the size of the data set to reduce compute times for initial training.
- If we are to use fastText for efficient text classification and representation learning, we will convert each review into a list of bigrams or trigrams and map these to word vectors.

Algorithms (From simple to complex): Logistic Regression, SVM, Naive Bayes, Random Forest (Balanced/Unbalanced) XGBoost, CatBoost. In general, we want to start with relatively simplistic algorithms, which are easier to evaluate. That way, we can easily identify sources of bias and variance, and adjust accordingly for more complex algorithms.

Hypotheses: Because this is an unbalanced dataset, as there are more positive reviews than negative ones. Therefore, the algorithms will see a lot more positive reviews, and be biased towards them as a result. Therefore, we need to find strategies that help mitigate the bias in the data, which will allow the algorithm to predict negative reviews better.

Evaluation: We want to evaluate the model based on its ability to correctly classify reviews. Simple accuracy metrics and structures like confusion matrices and ROC curves should be sufficient for this project

Timeline:

- Oct.6 - 19: Initial data cleaning/analysis, Logistic Regression, SVM, Naive Bayes
- Oct.22: Project Proposal Submission
- Oct.29: Project Proposal Presentation
- November: Complex Model Evaluation
- Dec.3&4: Final Presentation
- Dec 11: Final report