Michael Guo & Thomas Sieben
EECS 349 Project Proposal
November 5, 2018

What task will you address, and why is it interesting? You want to make this convincing, so spend about two paragraphs on this.

Have you ever been nervous submitting a comment on YouTube? The panic of not getting enough likes, being made fun of by your friends, or deleting a comment gone wrong? Well, we've been there. We will be constructing a ML model taking Youtube's greatest hits (the best comments out there) and showing you what you need to succeed in this day and age.

A user will be given a specific trending YouTube video and prompted to type a comment pertaining to the video. We will then predict the amount of Likes that the comment will receive. For example, for the video "***Bell's Theorem: The Quantum Venn Diagram Paradox***" the comment "***Today I learned I am not smart.***" received over ***1700 likes***! Based on our model, you should aim to get the highest number of likes, perhaps through humor relating to the video, a snarky remark, or just write a long rambling comment.

How will you acquire your data? This proposal is intended to serve as a sanity check that your project is doable -- so if you're inventing a new data set, be as specific as possible here.

We will be using the following datasets from Kaggle:

Trending YouTube Video Statistics and Comments
https://www.kaggle.com/datasnaek/youtube#USvideos.csv

Trending YouTube Video Statistics
https://www.kaggle.com/datasnaek/youtube-new#USvideos.csv

The dataset does an excellent job of capturing the information we're trying to use to train our ML model.

Which features/attributes will you use for your task?

We will be looking at the actual words of the comment itself to perform some sort of sentiment analysis, the number of likes a given comment has, the number of replies to a comment, the video title (from the video id).

One potential issue we have identified early is that a lot of comments with the most likes are comments from the original poster of the video (e.g., popular YouTubers). This might be difficult to filter out based on the format of our data but is something to keep in mind.

What will your initial approach be? I.e., what data pre-processing will you do, which ML techniques (decision trees, nearest neighbor, etc.) will you try first, and how will you evaluate your success?

We plan on using a naive bayes classifier to perform sentiment analysis on the corpus of words in the comments. For each video, it will have comments, and each video will have a unique corpus of words for the comments. We will weight the words based on the number of likes (scaling based on most number of likes for a comment and least number of likes) and the distribution of likes. For data preprocessing, we will need to match comments to a specific video_id in order to extract the video title for the comments we're analyzing.

Time permitting, we are also going to explore RNNs (neural nets) to generate a comment that is as popular as possible, creating it for the user (as opposed to the user typing their own comment).

Potential packages used:

- nltk
- numpy
- tensorflow