Analytic Plan

Ryan Waterman

Table of Contents

Business Objective

BetterHelp is an online therapy platform that has approached Twitter with a proposal to use its tweet data to train a sentiment analysis model. The model will be used in a targeted ad campaign that aims to integrate ads for their platform into the Twitter feed of users posting overwhelmingly negative tweets. The task at hand is a supervised classification task in which tweets with associated sentiment labels (0 for negative and 4 for positive) are used to train the model. Upon completion of the training and validation phase, this model can be deployed in an automated pipeline to scrape Twitter data and predict tweet sentiment. These predictions will be used in conjunction with BetterHelp's proprietary targeted ad algorithm to market to customers with high retention rates through therapy.

Dataset Overview

The target label for this dataset is 'sentiment', which is comprised of 0 for negative sentiment and 4 for positive sentiment. Note that this label is discrete, and the dataset will only include sentiments 0 and 4. The other critical feature in the dataset is the tweet text, captured in the 'text' feature. This is the data from which the target label is derived. Additional features in the dataset include "id", "date", "flag", and "user", but these features are auxiliary and will not be used when training the model. The dataset has been truncated from 1.6 million tweets down to 50 thousand to balance the sentiment class and reduce computational requirements for model training. The distribution of sentiment labels can be seen below.

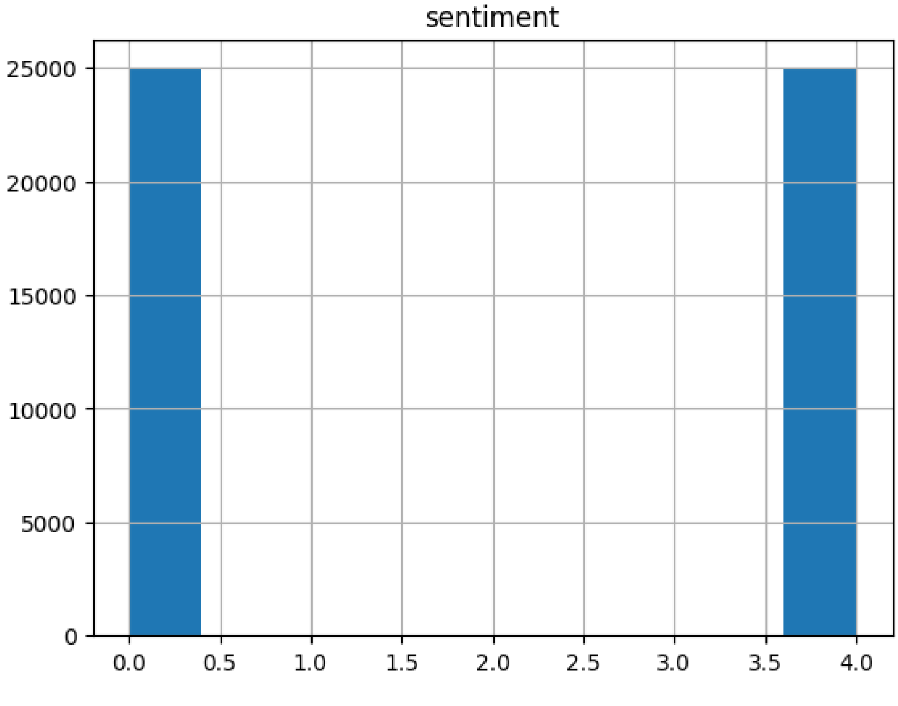

Image 1: Distribution of Sentiment Labels

Preprocessing and Feature Plan

Text data is inherently noisy and non-uniform; therefore, several processing methods will be applied to normalize the data and extract context and sentiment. Examples of noise in text data, particularly tweets, include emojis, twitter handles, URLs and HTML entities, abbreviations, and erroneous grammatical variances. The approach used to handle noise in this scenario is to remove all special characters, URLs, HTML entities, twitter handles, and emojis, and replace abbreviations with the equivalent dictionary word. Additionally, stop words will be removed to simplify the dataset and increase the frequency of words that are more influential on the tweet's sentiment. Note that the exploratory analysis did not include replacement of abbreviated words, as their impact on model performance will be evaluated during Phase 2.

Upon completion of data preprocessing, an analysis of token frequency and tweet length by sentiment was completed. Images 2 and 3 detail the distribution of the most frequent tokens in the preprocessed dataset.

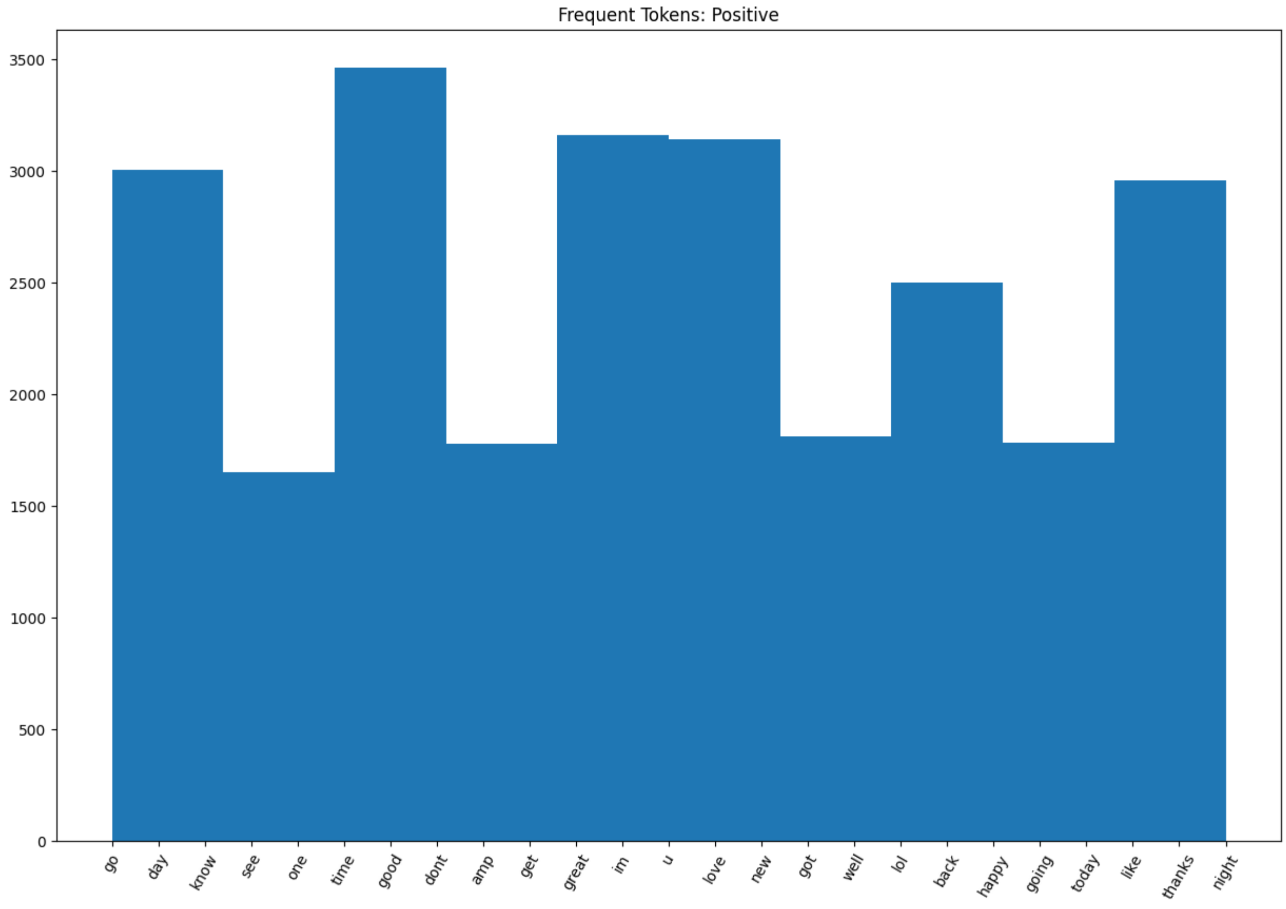

Image 2: Most Common Tokens in Positive Sentiment Tweets

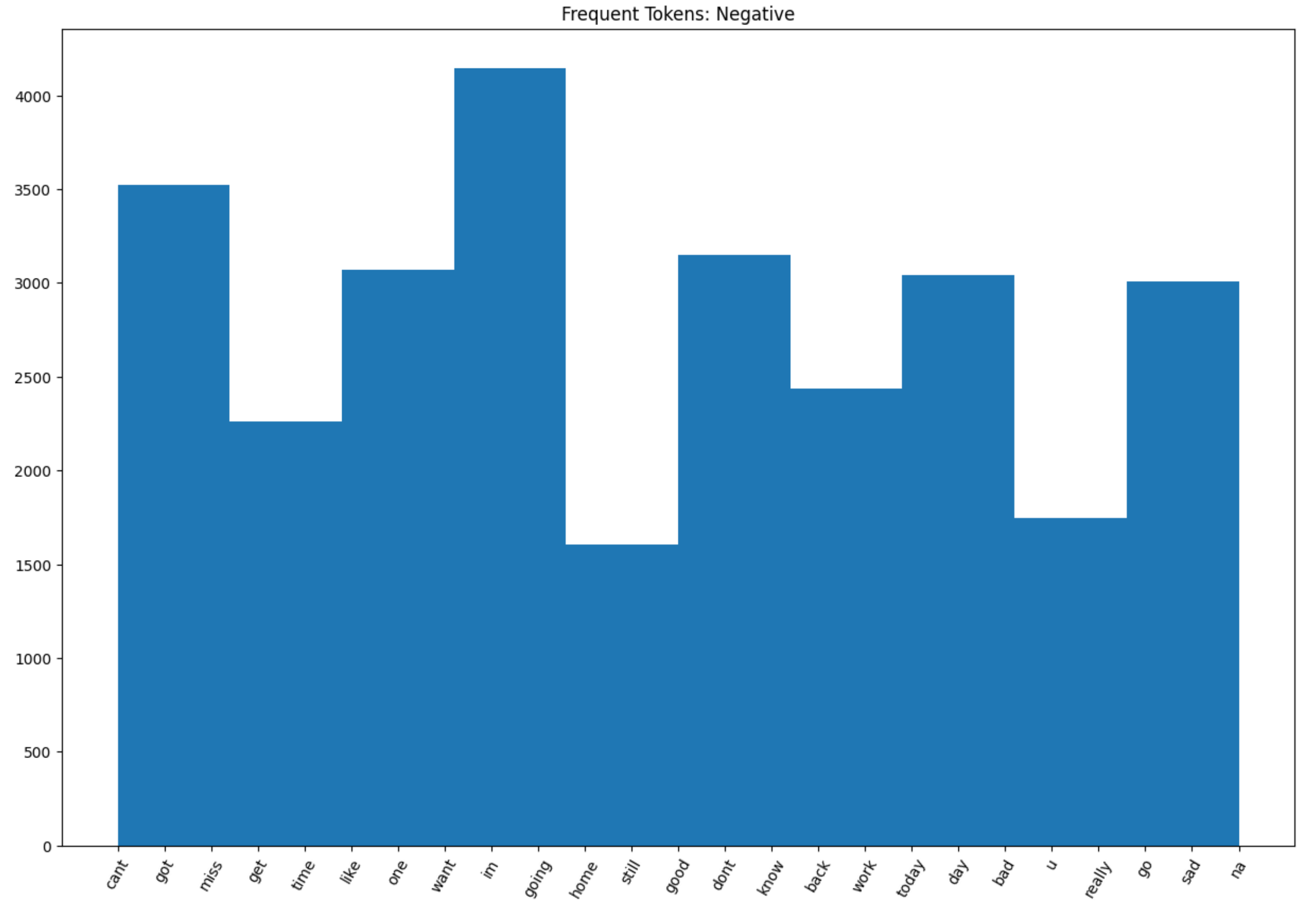

Image 3: Most Common Tokens in Negative Sentiment Tweets

There are some obvious trends in token frequency based on sentiment. For example, the most common token in the positive sentiment tweets is “good” and some of the most common negative sentiment tokens are “cant”, “bad”, and “sad.” Some notable tokens that are frequent in both sets of sentiment data are “im” and “u”, which would be removed during stop word removal if they were their grammatically correct, non-abbreviated counterparts. As mentioned prior, additional analysis is required to determine the impact of these abbreviations on model performance. Another feature of the tweet data to analyze is the length of the tweet. The distribution of tweet length (in words) by sentiment can be found below.

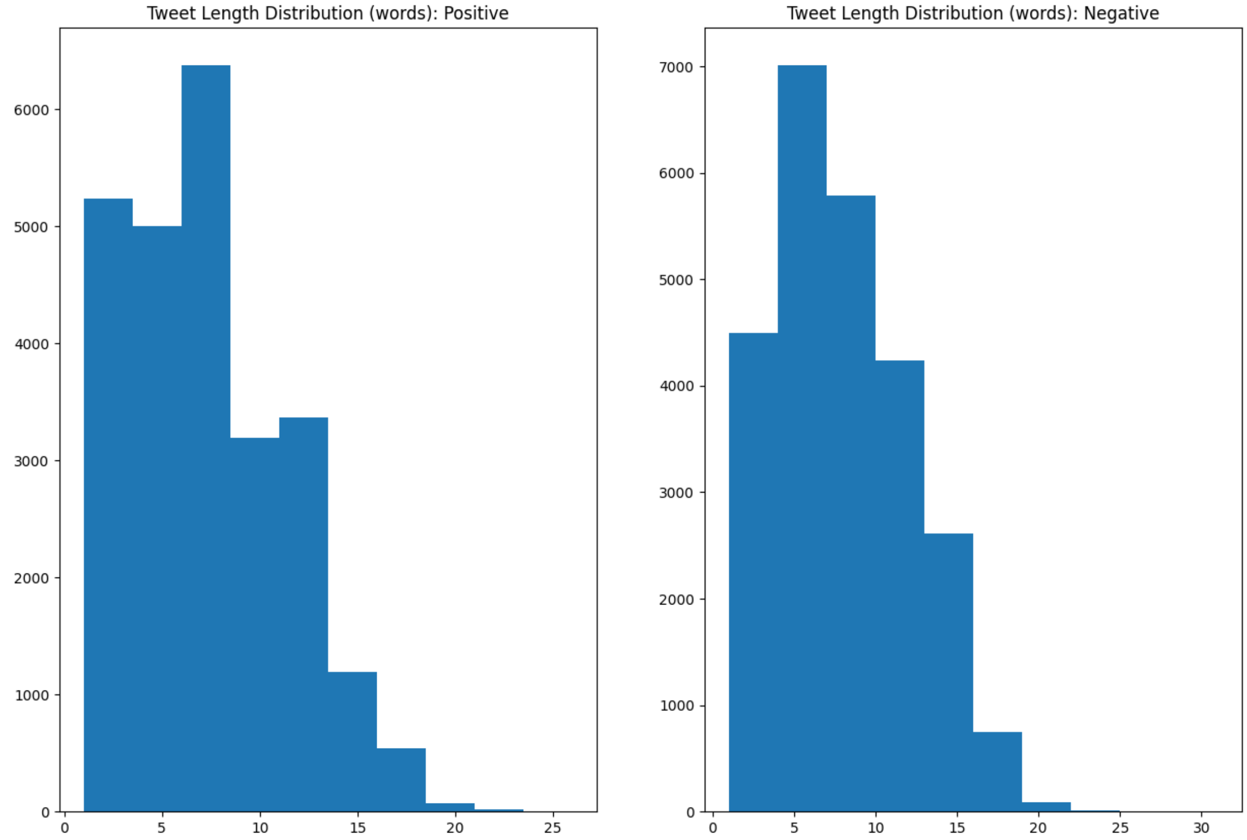

Image 4: Distribution of Tweet Length (in Words) by Sentiment

Interestingly, the positive tweets appear to be shorter on average than the negative tweets. The hypothesis for why this may be the case is as follows: positive declarations are more often accepted without the requirement for extra context, while negative statements often require evidence to support the claim, or there will be social repercussions due to ambiguity about the intent. An additional feature to predict tweet sentiment could be tweet length, something that will be experimented with during Phase 2.

Proposed Modeling Approach

Phase 2 will be highly analytical, relying on testing and experimentation to determine the optimal model architecture and feature set. The model architectures that will be explored during Phase 2 are Support Vector Machines, Decision Tree Classifiers, and a Deep Neural Network. Phase 3 will include more advanced deep learning architectures, such as the Advanced Transformer and DistilBERT. A cross-model analysis will be performed to determine the optimal architecture at each phase for both model accuracy and compute requirements.

Timeline and Risks

The timeline for the project is as follows:

- Week 1:
  - Complete feature engineering experimentation and optimize preprocessing pipeline.
  - Compare results of different vectorization methods, stemming, and lemmatization.
- Week 2:
  - Experiment with SVM, DTC, and custom DNN architectures to understand the effects of preprocessing choices and abbreviation substitution.
  - Identify the most effective architecture and the most critical features.
- Week 3:
  - Tune hyperparameters for optimal model selected in Week 2.
- Week 4:
  - Understand and experiment with implementation of advanced models.
- Week 5:
  - Test advanced model architectures on various preprocessing methods.
  - Tune hyperparameters for best performing model.
- Week 6:
  - Aggregate model performance data and report on findings.

Risks associated with this timeline are almost exclusively technical in nature. The PC that will be used for model training and testing contains an Nvidia 4080 Super GPU, which is CUDA enabled and can deliver approximately 50 TFLOPs. This should be sufficient for model training, but the performance will need to be monitored to ensure training does not take an excessive amount of time. Dimensionality of the data and model size can be reduced to mitigate this issue, which will require analysis to determine the trade-off of dimensionality and model complexity on performance. Additionally, there is some risk in the ability to implement the more complex model architectures, in which case fine tuning on trained models may be required.