May Paddor

MGSC 7650: Machine Learning

Individual Assignment Executive Summary

In this assignment, I created an online application called "Sentiment Analyzer" to showcase a sentiment analysis model to detect whether a patient's drug review is positive or negative. The dataset I used to train and test my model is the Drug Review Dataset on UCI Machine Learning Repository, and it includes three reviews: benefits, side effects, and overall comment.

For this assignment, I chose to use a TfidfVectorizer model because it can be more effective in sentiment analysis since it reduces the impact of common words. The other option I considered was a CountVectorizer, but I chose against it because it focuses on the frequency of tokens in each document instead of the importance of the tokens in the entire text. I decided to use a LinearSVC model because I felt the most comfortable performing a transformer-based model and utilizing the spacy package for transformer-based models. I tested five different tokenizers to tweak my model, and each tokenizer had different features from the Spacy and Afinn packages. Before I tweaked my model, I explored my dataset.

To start this assignment, I conducted an exploratory data analysis. I reviewed the data samples and created a histogram chart to determine whether the data was skewed. I found that the data was skewed towards positive reviews, with most reviews having a rating of 6 and up on a scale from 1-10. I used the spacy tokenizer we created for our text-based model class to determine how to split the data into positive and negative. Labeling 5 and below as negative gave an accuracy and F1 score of <70, marking 4 and below as negative gave an accuracy & F1 score of ~77, and labeling 3 and below as negative gave accuracy and F1 score of ~80. To validate the accuracy of these data segments, I used the test data to test the accuracy and F1 scores. I received the highest accuracy when I labeled the data with ratings 4 and below as negative.

Next, I created my five tokenizers. The first tokenizer I used performed lemmatization, stemming, and removing stop words. The second tokenizer I used focused on using words from the Spacy NLP library, which makes it more efficient when analyzing large portions of text. The third tokenizer I used conducted lemmatization, utilized the Spacy NLP library, and specifically examined the nouns and adjectives in the function. The fourth tokenizer I used was like the third tokenizer but included stop words, verbs, and adverbs. Finally, the last tokenizer I tested incorporated aspects of the fourth tokenizer and utilized Afinn lexicon, which assigns numerical scores to each word in the lexicon based on perceived sentiment. I evaluated the tokenizers by looking at the test data scores because the model could overfit the training data. The second tokenizer was the most effective, with an accuracy score of 90.15 and an F1 score of 94.82 based on positive labeling accuracy.

After creating my model, I made a Streamlit app to showcase the insights I found with the data and the steps I took to develop and refine my model. I also included a section for users to download reviews to test the sentiment analysis function.