Sentiment Analysis for NLP

Joseph T. Tustin

/github.com/joetustin
/in/joseph-tustin/
joetustin67@gmail.com

Intro

The Rotten Tomatoes Movie Review Dataset consists of over 420,000 reviews which are in the form of simple sentences. These reviews are classified as either positive or negative. The purpose of this capstone project is to to build a model to analyze the text to predict the review. With random guessing, we would have a baseline predictive ability of 50%.

Problem Statement

Natural Language Processing is hard! Words (unigrams) have meaning. Combination of words (bigrams and trigrams) can hold even more meaning However, despite all this information that words and combinations of words hold, there can still be double meanings, sarcasm, and contradictory text. It is estimated that given some random text only 65% of all people will agree on its given meaning. Despite these limits of interpretability, sentiment analysis has become big business in data science. Even small gains in accuracy can lead to big gains in sales.

Positive Review
"Manakamana doesn't answer any questions, yet makes its point: Nepal, like the rest of our planet, is a picturesque but far from peaceable kingdom."

Positive Review
"Wilfully offensive and powered by a chest-thumping machismo, but it's good clean fun."

Negative Review
"It would be difficult to imagine material more wrong for Spade than Lost & Found."

Negative Review
"Despite the gusto its star brings to the role, it's hard to ride shotgun on Hector's voyage of discovery."

Text Cleaning

Text cleaning is an important first step in natural language processing. After text cleaning, the simplified words in the reviews must be turned into a numerical matrix using count vectorization as well as tf-idf factorization. The typical work flow for text cleaning is show below:

Input Text Documents
↓
RegEx: to remove num, punc., char
↓
Tokenize: Sentence to list of words
↓
Stopwords: Remove common words
↓
Lemmatize to find common basis

Once this process has been completed, the data from the previously listed reviews is shown below.

['manakamana', 'answer', 'question', 'yet', …]
['wilfully', 'offensive', 'powered', 'chest', …….]
['would', 'difficult', 'imagine', 'material', 'wrong',..]
['despite', 'gusto', 'star', 'brings', 'role', 'hard',..]

As a picture can be worth a thousand words, the word cloud shown represents the frequency of the most common words as seen in the positive reviews. Words, such as "best", "love", and "great" stand out as being associated with positive reviews.

Word Cloud for Positive Reviews

Methodology

With the normalized, numerical matrix representing the words in each of the reviews, models were created with a small subset of the data using Multinomial Naive Bayes, Random Forest, and Logistic Regression to find the best predictive model while hyper tuning the model parameters using gridsearch with cross-validation.

	Training Accuracy Score	Test Accuracy Score
Multinomial Naive Bayes	0.78	0.73
Loigistic Regression	0.80	0.70
Random Forest	0.65	0.62

Once the best model (Multinomial Naive Bayes) was chosen, the model was scaled up to work on a larger dataset using an AWS m5 instance. With the finalized model, a flask app was created to allow users to interact with the model.

Results

Put your information here. Remember to size your font accordingly.

Test Accuracy vs Training Data Size

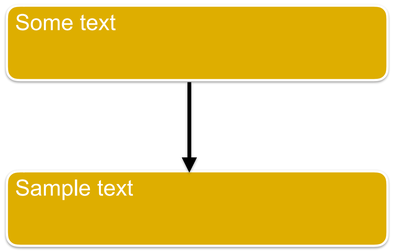
	Predicted Negative Review	Predicted Positive Review
Actual Negative Review	1796	670
Actual Positive Review	677	1857

Would like to add text box showing four examples of mis-classified text

Would like to show two lists of words representing the highest probability of determining a positive or negative class….

Future Work

Great progress has been made in the area of natural language processing using Neural Networks as evidenced by the success of the algorithm's such as "Elmo" and "Bert".. These neural networks are pre-trained on large text datasets such as wikipedia and then can be reapplied to other text datasets. I plan to use these models in the future to achieve accuracy scores which should exceed 90%. While quite effective, these models still suffer from the loss of interpretability imbedded in the deep calculations of the weights.


Some text
Sample text