**Natural Language Understanding**

**Sentiment Analysis**

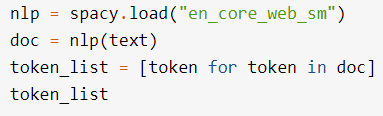
Mert Akkor

In this project, I made a sentiment analysis on Movie Reviews and classify the review whether it is positive or negative. To touch the sentiment analysis topic, it is a powerful tool that allows computers to understand the underlying subjective tone of a piece of writing.

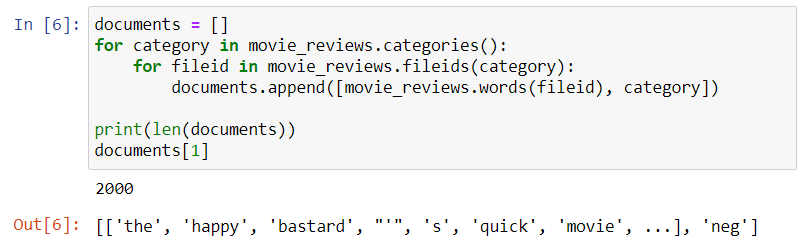
Throughout the process of the project, I followed the following steps;

* Tokenizing sentences
* Removing stop words
* Normalizing words (tried stemming but did not apply)
* Vectorizing text (Feature sets)

Firstly, I used NLTK’s Movie Review library to work on this project. It has mixed positive and negative 2000 movie reviews. As we have done many times in our labs and also assignments, ‘movie\_reviews.words()’ makes the tokenization itself for us. Inside of it, it makes the same thing we have done:



As a result of its use, I got my words in the reviews as a tokens.

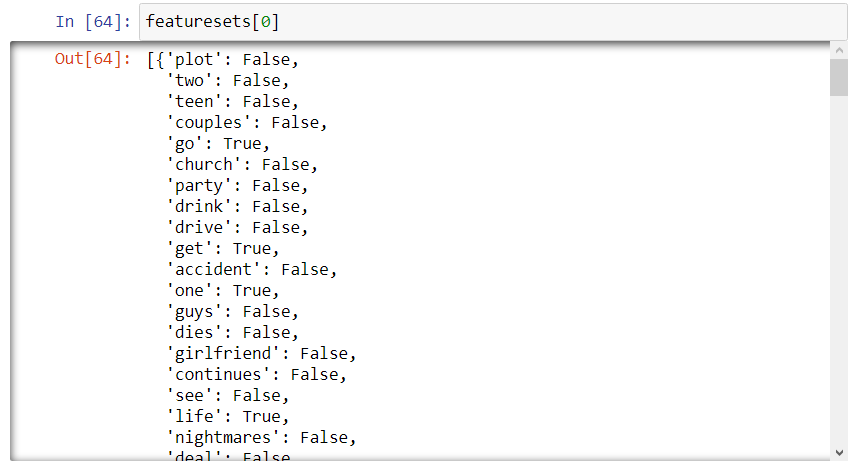


After achieving my tokens, second step is to get rid of stop words. Stop words are words that may be important in human communication but are of little value for machines. This makes big problems and lead to misclassifications. Luckily, spaCy comes with a default list of stop words. Also, for my case, I added a custom stop list as well, because in reviews there are many punctuations which are not necessary in categorizing positive and negative.

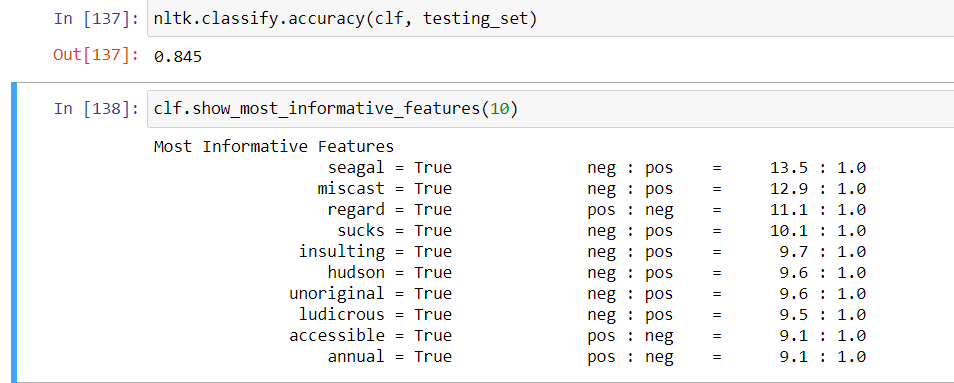
After removing stop words, I tried stemming as well in the sense of normalizing words and to increase my accuracy and get better results, however, I got meaningless words like ‘movi, charact, inept, etc.’ so did not apply stemming for improvement. Below, you can see the application of removing stop words and most common 10 words.



Lastly, similar to vectorization, I created feature sets for each category (positive, negative) and used their features to train and test. In feature sets, we have ‘True’ and ‘False’ labels. ‘True’ is assigned if word in feature set can also be found in review, otherwise ‘False’. By this way, we train which word contributes to positive reviews and which contributes to negative reviews.

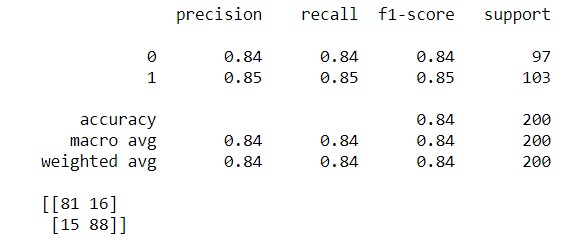


For training, I used NLTK’s Naïve Bayes Classifier, also first 1800 reviews are used for training and last 200 is used for testing. I got accuracy of 0.845.



As a sidenote, since I make shuffling earlier, these may change after each run.

For my evaluation, I got help from sklearn’s classification\_report function since I find it much easier and more efficient than NLTK. So as a result, I got;



To conclude, in this project, with the knowledge we learnt from our labs and previous assignments, I tried to implement a sentiment analysis on NTLK’s movie reviews library and classify the positive reviews and negative reviews with the accuracy of 0.845 on NLTK accuracy and 0.84 on sklearn F1-score.