Project 2: Supervised Learning

How to Run:

The main run file is code/naivebayes.py. Unzip everything and run the "naivebayes.py" file. The output is the first element of bag of words example, a prediction for a test tweet, a list of output of the test file and accuracy.

Method used in the main file:

First the lite.csv file was read. The file contained data, user, text, polarity. A bag of words was created from the text for each line by list_from_tweet() function. The text was preprcessed by removing links, usernames, stop words, all the non-alphabets and then lemmatized.

Naïve Bayes function was used for the supervise learning. The Prior probabilities were found for positive and negative tweets separately. Then the list of tuple that was got from previous method was passed to find the conditional probability. As the training, dictionary for all positive words and negative words was created. The key was the word and the the probability was the value.

The formula used is.: $P(Word|polarity) = \frac{(Word\ frequency+1)}{(len(polar\ words)+len(all\ unique\ words))}$

For predicting a tweet was converted in the bag of words and the P(Word | polarity) was extracted by checking it in dictionary. We found out the probability for both polarity (i.e positive and negative.) using following formula.

$$P(polarity|tweet) = P(Prior\ polar\ probability\ ) * \prod P(Word|positive)$$

These probabilities were compare and the polarity with larger probability was taken as the polarity of that tweet.

For getting the Accuracy: the model was tested for sample of the known data. We tested it in 1250 rows of data. It was repeated for different data and all had closed range accuracy. From the final data point the accuracy was found to be 59%. The neutral part was omitted in this process because there were only 1 and 5 in training data/testing data.

For Bonus: Some improvements in the model was by going the weight to positive and number. The words were checked if it was in file with positive of negative words. If it was found, it's probability value was multiplied by 2. It was done so that when predicted the positive words would have more influence in the final probability.

Additionally, while preprocessing the words with occurrence smaller than 5 were not counted. It was done to remove the noise from the data. The misspelled words or names were omitted this way. The preprocessing process of lemmatization and removing stop words also reduce the word count reducing the training time.

Output.csv file contains the prediction of this model including the neutral. For this the ratio of probability of positive to probability of negative was taken and if it was close to 1 then it was labeled as neutral(3). It also contained output_without_neutral.csv which exclude the polarity of neutral.