Deception detection assignment report:

I have implemented a simple Naïve Bayes classifier with some extra features. My code consists of 2 functions, namely, get_data and get_count.

In get_data() function, any given dataset can be parsed through it (training or test data-set) to obtain a list containing the ID of the review as the first element and the modified scanned word list as the second element. The modified word list is extracted in the following method,

- Removed all the punctuations
- Removed all the '\n' terms
- Split the sentence with space to obtain the words
- Using regex found all the exclamation marks, question marks or bunch of them appearing together in the given training data
- Added a string to the scanned word list stating the presence of the above punctuation marks like PUNCxEXCLAMATION_POINT, PUNCxQUESTION_MARK, PUNCxINTERROBANG respectively.

Parsed the True training set and False training set separately to the function and obtained the lists accordingly.

With get_count function, the count of each word in the lists of the respective classes are computed, to have vocabulary filled with frequencies for calculating the probabilities corresponding to the classes.

Prior_probabilities of both true and false reviews are computed as,

⇨ prior_T = Number of true reviews/ total number of reviews

⇨ prior_F = Number of false reviews/ total number of reviews

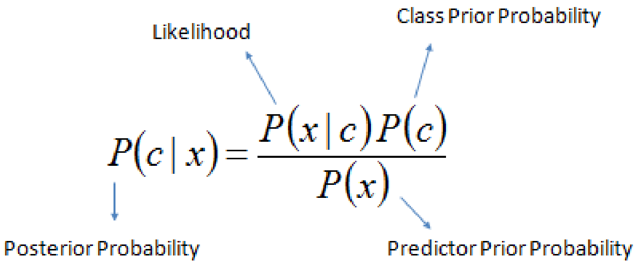
Removed certain chosen stop-words from the scanned lists. These words are some of the words with high frequency that doesn't support to the sentiment of the review.

The final part is finding the likelihood of each word in the vocabulary to each of the classes. Since some words in the vocabulary are absent in the one of the classes, Laplace smoothing is done to avoid discrepancies. The likelihood is calculated as follows,

$$\hat{P}(w_i|c) = \frac{count(w_i,c)+1}{\sum_{w\in V}(count(w,c)+1)} = \frac{count(w_i,c)+1}{\left(\sum_{w\in V} count(w,c)\right)+|V|}$$

Where |V| is the size of the vocabulary.

To detect the deception of a new review, the following formula is implemented as it is, using the above computed probabilities. The words that are unknown in the new review based on our vocabulary is omitted,



$$P(c \mid x) = \frac{P(x \mid c)\,P(c)}{P(x)}$$

$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

The product of the likelihood of the words with respect to each of the class and prior probability of the respective classes are found out and the class of the maximum result among them is assigned to a new review from the test-set (whose data is already parsed through the get_data() function) and written to a text file with its ID.