Connor Frazier
Natural Language Processing
Homework 1
1/29/20

PS1_Reviews:

Basic Features:

In createBasicFeatures, the goal was to create sparse matrix where every word is a feature represented by a column and every review is represented by a row. The values of the matrix are the counts of the words in the review. For example, if the i-th column word is “great” and there are four occurrences in the first review, the matrix value for the first wow at the i-th column would be four. To create this matrix, the method first loops through each of the texts. Inside the loop, the class label is add to the classes list, and the text is added is added to the texts list. The text is cleaned before adding which includes removing digits, underscores, and punctuation leaving only the words with alphabetic characters. After these steps, the texts are now in a list and passed to the vectorizer which creates the sparse matrix. Finally, the vocab list is obtained from the vectorizer. Lastly, the texts(matrix), classes(list of classes for each sample), and vocab (list of every word in the corpus) are returned.

Fancy Features:

For the fancy features method, I focused on creating features that would allow the classifier to find the more descriptive words in the texts and find the differences in language used in both types of reviews. To do this I focused on representing the reviews by their words’ “polarity scores”. A ratio I came up with to represent how much value a word has; for example, a word that occurs in more positive reviews than in negative reviews would have a higher ratio than a word that occurs the same amount in both classes. By doing this, the classifier was able to avoid being misled by common words.

To find the polarizing words, I created a dictionary that would hold the counts of each words for the number of occurrences of the word in positive and negative reviews. These counts were obtained by looping through all words and reviews in the same way as the basic

features method. The polarity score of each word is calculated by dividing the larger class count by the smaller class count. The sparse matrix is then created by looping through the reviews again, and creating a row for each review and a column for every word in the vocabulary. For each word in the review, the words column value for the row is the polarity score of the word from the dictionary. Finally, the vocabulary of the list is obtained from the dictionary. Lastly, the texts(the matrix created), classes(list of classes for each sample), and vocab (list of every word in the corpus) are returned.

The accuracy of the classifier did improve by using the polarity scores instead of the counts of the words. I believe this to be the fact that the polarity score is more meaningful than the count of the word because it represents value of the word to its respective class. The polarity score value is much more representative of the value of the word when deciding the sentiment of the class when compared to how many times a word is used.

Note: The evaluation using this method takes between five and ten minutes (on my computer) due to the number of calculations.

PS1_Shakespeare:

Basic Features Findings:

For the implementation of createBasicFeatures in this file, please refer to the Basic Features section of the PS1_Reviews section of this document. The only change is the classes list has become the genres list.

The features that the classifier assigns high weights do not tell us much about comedies and tragedies. There is no obvious theme to the words for the comedy or tragedy genre meaning that classifier never finds the difference between the two classes. The classifier makes a few mistakes; first, it gives high weights to common words like “you”, “i”, “his”, “and”, and “him”. These words are common word that don’t show a difference between the two

classes. The second mistake the classifier makes is that it is not able to distinguish between the different forms of the same word especially in the L1 norm version, meaning that the classifier is missing more important features. Lastly, in both norms and especially the L2 norm, a good portion of the words with high weights are proper nouns from the plays, which don't tell much about the genre of the play as they are just he just the subject of the play. These all points to the fact that the current set of features is not good enough for the classifier to choose between the genres. These mistakes lead me to believe that the classifier needs a way to find the types of languages used by the different genres instead of the just the words used.

Interesting Features:

Since the classifier using the basic features for the Shakespeare plays was having a problem with assigning high weights to low value words. My goal was to represent the features of the play in a way for the classifier to focus on more high value words with respect to the classification task. The first part was to calculate the polarity score for each word in the same way as the fancy features in the Reviews file. The second part is to get the part of speech for every word in the vocabulary using the nltk library.

After obtaining this data, the vocab list is created by adding all words that are not nouns and that have a polarity score greater than 2. The texts are then passed to binary vectorizer with the vocab list to create the matrix. Where the values for each row and column are one if the play contains the column word and zero if the play does not contain the column word. These features greatly improved the performance of the classifier.

There are two affects that these features had; the first is that by using the polarity score to choose words, the common words between the the two genres were filtered out which allowed the classifier to better learn the difference between the language of the plays. This also reduced the affect of different forms of words because words in their basic form seemed more likely to occur. The removing of the nouns was a way to remove the topics(subjects) of the plays. While they are different across the the genres, they are are also different in all of the

plays and therefore do not matter much in the classification task of the genres. After these features were implemented, the high weight words resembled the difference in language between the genres and therefore the better classification accuracy.