Blake Johnson
9663980

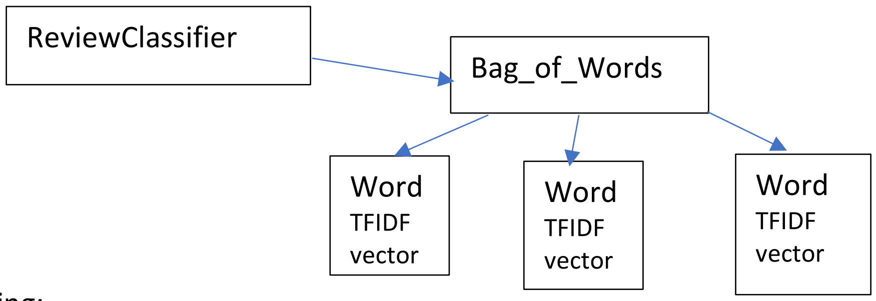
MP1 Report

Overview:

My project for MP1 consists of multiple classes, that interact with each other to stay modularized and simplify the debugging and organization. My top level class is called ReviewClassifier. This class has two bags, one with the learned positive data, and the other with the learned negative data. My medium level class is called Bag_of_Words. This class has a few attributes, but mainly consists of a dictionary called wordlist. WordList being a dictionary is very important because it allows us to store and lookup values with a run-time of O(1). This keeps our learning and classification quick. With such a large dataset required to train, and a large data set to classify, it is crucial that the run-time of my program is quick, so that I can identify problems and debug them efficiently. My lowest level class is the Word class. This class contains almost all of the statistical information required for the extractors and classifiers.

Python Version: Built with python 3.7.0 but should also work with 2.7

Class Diagram:



Parsing:

The first, and arguably most important task of this project was parsing. With incorrect parsing, this project will be extremely frustrating and time consuming, as I learned from experience. However, struggling with removing empty character space ‘ ‘, I was able to parse the reviews correctly, breaking at each <br /><br />. I also removed ***stop words*** by creating a file full of stop words, and using each word in the file to compare to the data in the training and test sets.

Feature Extractors:

Bag of Words extractor was implemented using dictionaries for their constant look ups and inserts to keep track of the number of appearances of each word and which reviews they appeared in. The bag of words extractor is represented as an instance of the Bag_of_Words class.

Term frequency – inverse document frequency was implemented using a slightly different approach than the bag of words. In order to calculate the tf-idf of each unique word in each review, we needed to know the number of documents the word appeared in, meaning we had to iterate through the entirety of the reviews twice. This caused enormous run-time complexity issues, but after some tweaking, I figured out that I needed to create a dictionary and iterate through the reviews to store this value before doing my learning, allowing me to have a much faster run time. The TF-IDF of each unique word is stored in the "self.statsListTotal" attribute of my Word class.

Classifiers:

Gaussian Naïve Bayes classifier was used in conjunction with both the TF-IDF extractor as well as the Bag of Words extractor to classify the data using a Gaussian distribution to represent the probability of a word given a class type (positive or negative). This classifier uses numpy to obtain continuous parameters from our extractors and their corresponding data sets.

Multinomial Naïve Bayes classifier was used in conjunction with only the Bag of Words extractors to calculate the conditional probability of each review given a class (positive or negative).

Results:

```
*******Multinomial Bag of Words Results*******

#### POSITIVE TEST ####
###### MULTINOMIAL BOW ######
Positive Reviews: 2192 76.27000695894223%.
Negative Reviews: 682 23.72999304105776%.

#### NEGATIVE TEST ####
###### MULTINOMIAL BOW ######
Positive Reviews: 781 24.984005118362123%.
Negative Reviews: 2345 75.01599488163787%.

*******Gaussian Bag of Words Results*******

#### POSITIVE TEST ####
###### GAUSSIAN BOW ######
Positive Reviews: 1832 63.766098155238424%.
Negative Reviews: 1041 36.233901844761576%.

#### NEGATIVE TEST ####
###### GAUSSIAN BOW ######
Positive Reviews: 1294 41.394753678822774%.
Negative Reviews: 1832 58.60524632117722%.

*******Gaussian Tf-Idf Results*******
```

```
#### POSITIVE TEST ####
###### GAUSSIAN TFIDF ######
Positive Reviews: 1631 56.76992690567351%.
Negative Reviews: 1242 43.230073094326485%.

#### NEGATIVE TEST ####
###### GAUSSIAN TFIDF ######
Positive Reviews: 1503 48.0806142034549%.
Negative Reviews: 1623 51.9193857965451%.
```

Collaborators:

All work was done individually, however I discussed concepts of the project with Fernando Mendoza and David Roster.