**Document Splitter**

**Description:**

This process is used to split the files with one review per line to files with one word per line. This is the format that is needed for the TF-IDF code to work.

**Input:**

The input for this process is five files, one for 1-star reviews (Score1.txt), one for 2-star reviews (Score2.txt), one for 3-star reviews (Score3.txt), one for 4-star reviews (Score4.txt), and finally one for 5-star reviews (Score5.txt). Each one of this file contains a review per line.

**Output:**

The output is five files called Score1-split.txt, Score2-split.txt, Score3-split.txt, Score4-split.txt and, Score5-split.txt. Each one of this file contain the same reviews as the files in the input but only one word per line.

**TF-IDF**

**Description:**

Computes the TF-IDF score of every word in every document

**Input:**

Five files called Score1-split.txt, Score2-split.txt, Score3-split.txt, Score4-split.txt and, Score5-split.txt that contain every review in their class but only one word per line.

**Output:**

Five files called Score1-tfidf.txt, Score2-tfidf.txt, Score3-tfidf.txt, Score4-tfidf.txt and, Score5-tfidf.txt that contain the tfidf score of every word in their class.

**Naïve Bayes**

Description:

Matlab script that creates reads the training data, training labels, test data, and test labels. Also, it creates Naïve Bayes classifier using the training data and training labels. Then, it uses the classifier with the test data to predict the classes. Finally, it creates a confusion matrix that is used to gauge the performance of the classifier.

Input:

Four files: #trainingData.txt, #traininglabels, #testData and #testLabels where “#” is a number from 1 to 5. The number 1 files contain the data for the top 10% most important features. The number 2 files contain the top 30 most important features. The number 3 files contain the top 100 most important features. The number 4 files contain the top 1% most important features. Finally, the number 5 files contain the top 20% most important features. The names of this files have to be changed on the matlab script depending on which data you which to run.

Output:

* A (number of reviews) x (number of features) matrix called predicted labels with the predicted class for each member in test data.
* A 5 x 5 matrix called confusion matrix that contains the confusion matrix for the data that was classified