## 20 Newsgroup Document Classification Report

## Problem Statement

This project will focus on classifying the 20 news data using machine learning algorithms such as Naïve bayes, Random forest, SVM etc. It is a supervised classification problem, there are news of 20 categories where each piece of news belongs to one category.

The goal is to extract proper features and build an effective model to assign each piece of news to the correct category.

I will explore the dataset in the beginning on the training part by extracting useful keywords and build vectors of features from the texts of news. Then I will use several classification methods to do classification based upon those vectors and then compare the efficiency of these classifiers on the test data and choose one. I will also use gridsearch to find the best parameters for classifiers.

## Dataset

I will apply classification algorithms on 20 newsgroup dataset from CMU Text Learning Group Data Archives. It has a collection of 20,000 messages, collected from 20 different newsgroups. The news will be classified according to their contents

The data is organized into 20 different newsgroups, each corresponding to a different topic. Some of the newsgroups are very closely related to each other (e.g. comp.sys.ibm.pc.hardware / comp.sys.mac.hardware), while others are highly unrelated (e.g misc.forsale / soc.religion.christian). Here is a list of the 20 newsgroups,

|  |  |  |
| --- | --- | --- |
| comp.graphics | rec.autos | sci.crypt |
| comp.os.ms-windows.misc | rec.motorcycles | sci.electronics |
| comp.sys.ibm.pc.hardware | rec.sport.baseball | sci.med |
| comp.sys.mac.hardware | rec.sport.hockey | sci.space |
| comp.windows.x |  |  |
|  | talk.politics.misc | talk.religion.misc |
| misc.forsale | talk.politics.guns | alt.atheism |
|  | talk.politics.mideast | soc.religion.christian |

20news-bydate.tar.gz unpacks into two top-level directories: 20news-bydate-train and 20news-bydate-test. Both of these contain 20 subdirectories of newsgroup data, where the directory name is the same as the newsgroup name, and each of these contains a set of newsgroup messages, roughly 400—600 posts for the training sets and roughly 250—400 posts for the test sets.

## Analysis

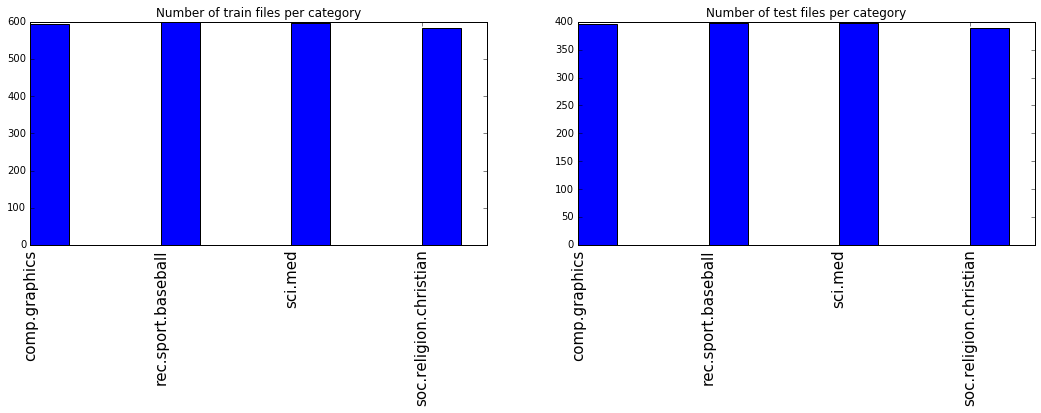
In this project, I have taken only 4 classes for data exploration and visualization, mainly, **['comp.graphics','rec.sport.baseball','sci.med','soc.religion.christian'].** There are 2374 samples in training set and 1580 in testing set and each sample is labeled.

## Similar analysis can be done for other classes. The Classification shall be carried over both 4 classes and 20 classes.

## Data Exploration and Visualization

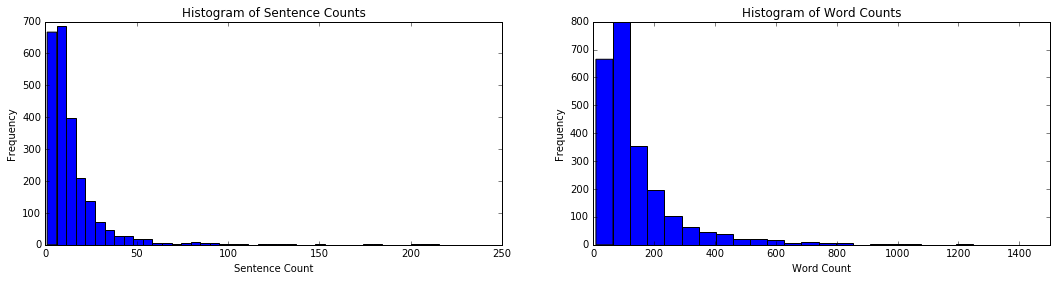
I checked some stats for corpus through barplots, histograms:

1. Below is distribution of train and test files per class. It can be seen that each class has almost same number of files and thus data seems to be balance.



1. Histogram of Sentence and Word counts.

Below is histogram of sentence and word counts of training data.

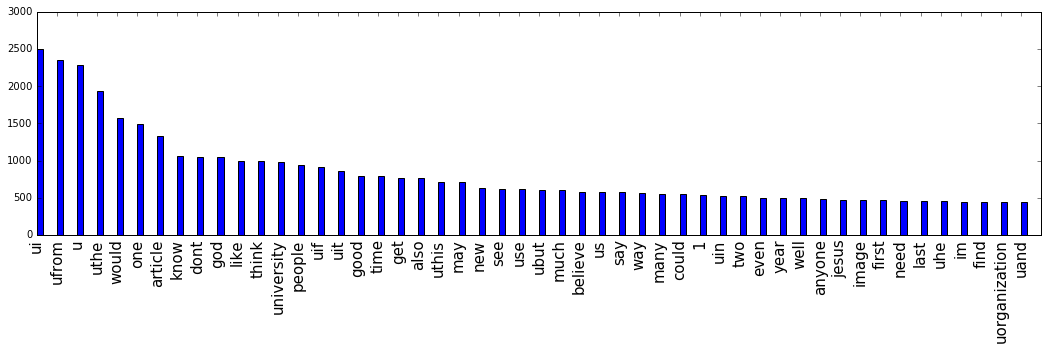


I have used nltk library for sentence and word tokenization. I have removed the punctuation and removed the stop words.

It can be seen that most files have less than 40 sentences and median is close to 20. Also most files have less than 400 words with max word count of 5613 and minimum word count of 9. (Please note that this is after removing the punctuation and stop words)

1. Histogram of top 50 frequent words

This is plotted across full training data.



The total number of unique words after removing stop words are 73534. This is a huge feature set as compared to number of sample in training data. I am going to reduce this number by setting parameters while doing vectorization of text.

**Classification Algorithms and Methodology**

**Data Preprocessing**

First of all, I extract features from the news data. These features can be words, sentences or phrases or combination of them all. I have considered using single words as features allowing for model complexity. I have used Tf-Idf algorithm to extract features rather than simple word frequencies. The tf-idf value increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general. This problem has over 2000 training samples with four categories, a multi-class classification problem with high dimension.

After removing the punctuation, turned letter into lowercase and removing the stopwords, there were a total number of 73534 words. I further reduced this feature set by setting the min\_df parameter to 5 in Tfidfvectorizer. This ignored terms that had a document frequency strictly lower than 5 while building the vocabulary.

The final feature vector was (2374, 8521) –

<2374x8521 sparse matrix of type '<type 'numpy.float64'>'

with 183053 stored elements in Compressed Sparse Row format>

**Classifiers**

I have used Python scikit-learn package to do the feature extraction, as well as Naive Bayes

Learning, Decision Tree learning and SVM. I have used Naïve bayes as my benchmark classifier. The parameters of NB and Decision Tree were default. I imported

Multinomial Naive Bayes classifer from sklearn.naive\_bayes, trained them on the extracted

feature matrix and predicted the categories on testing feature matrix.

**Benchmark Classifier – Naïve Bayes**

The accuracy of NB classifier was 94.81%

Training data Accuracy: 98.652064027 %

Test data Accuracy: 94.8101265823 %

It looks like NB model performed well in classifying the 4 classes.