**The University of Texas at Arlington**

**Department of Computer Science and Engineering**

**Spring 2017**

**Data Mining**

(CSE-5334)

**Assignment 5**

**Submitted By –**

**Prajal Mishra [1001434611]**

**Naïve Bayes classification**

**Introduction**

We are asked to implement naïve bayes classification method in python for text classification of text articles. The data set that we used is based on a from the abstract and introduction of scientific articles that come from three different domains:

1. PLoS Computational Biology (PLOS)

2. The machine learning repository on arXiv (ARXIV)

3. The psychology journal Judgment and Decision Making (JDM)

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a feature in a class is unrelated to the presence of any other feature.

**Algorithm**

First, we split the data into two folders, training and testing and its done using a coin toss algorithm to increase the randomness and effectiveness of the algorithm. Then we preprocess the data in which we form a vocabulary with stopwords removed. Here, stopwords are common uninformative words listed in the file stoplist.txt. Feature set is generated whose size is one more as it contains the class label. Then we use the training vocab and classify the data and based on the classification accuracy of the naïve bayes classifier is calculated.

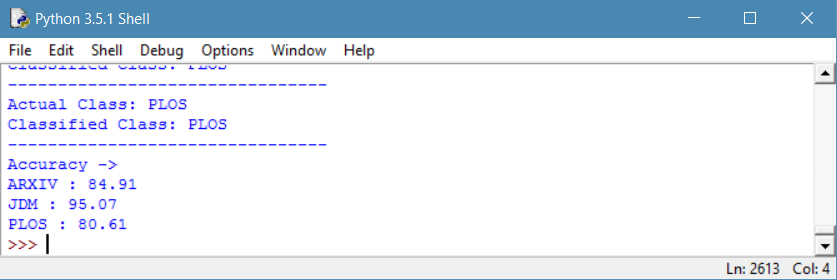
**Problems Faced**

There were several problems faced during the programming and execution phase of the program. While dividing the files, due to the random coin toss algorithm, no fixed size of the length of the data vector could be determined because of which it was getting very difficult to work on the limits of the loop.

Also, while classifying the data, due to the issue of random coin toss algorithm output for dividing the files, it was very difficult to have a control on the classification taking in consideration the random sizes.

The time taken to compilation is always a lot, so every time I make a change, I had to wait for several minutes to check the output of the program.

**Output Screen**

****