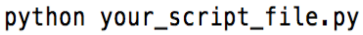
CSE 5334 - DATA MINING

Spring 2017 – Assignment 5
Due: 11:59pm Central Time, Tuesday April 11, 2017

REQUIREMENTS:

Read the following requirements carefully, and make sure you follow every rule. If you fail to meet any of requirements, marks will be deducted accordingly.

- The project document describes how you tackle the problem
  - No limitation on the number of pages, or the format of the document;
  - The document should clearly describe how you design and implement. **(5 points)**
- Submit your source code and project document
  - ONE zip file contains all source codes and the project document. **(5 points)**
  - You must implement the whole project by yourself. Academic dishonesty will have serious consequences.
  - Your source code must pass compilation. Any non-executable submission is not acceptable.
- Execution
  - Your program has to be executed using command line as followed: **(5 points)**



```
python your_script_file.py
```

THE PROBLEM:

In this assignment, you will implement the Naïve Bayes classification method and use it for text classification of scientific articles. The data set you will use contains sentences 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)

Each domain contains 300 articles (again, only the sentences from the abstract and introduction). You need to split the data into training (50%) and testing (50%) sets.

TASKS :

Your code should accomplish the following tasks:

1) **Pre-processing step:**
   This first step converts scientific articles into features to be used by a Naive Bayes classifier. You will be using the bag of words approach. The following steps outline the process involved:

a. Form the vocabulary. The vocabulary consists of the set of all the words that are in the training data with stop words removed (stop words are common, uninformative words such as "a" and "the" that are listed in the file stoplist.txt). The vocabulary will now be the features of your training data. Keep the vocabulary in alphabetical order. **(15 points)**

b. Now, convert the training data into a set of features. Let M be the size of your vocabulary. For each article, you will convert it into a feature vector of size M+1. Each slot in that feature vector takes the value of 0 or 1. For the first M slots, if the ith slot is 1, it means that the ith word in the vocabulary is present in the fortune cookie message; otherwise, if it is 0, then the ith word is not present in the message. Most of the first M feature vector slots will be 0. Since you are keeping the vocabulary in alphabetical order, the first feature will be the first word alphabetically in the vocabulary. The (M+1)th slot corresponds to the class label. An A in this slot means the article is from class “arxiv” while a J in this slot means the article is from class “jdm” and a P in this slot means the article is from class “plos”. **(15 points)**

2) **Classification step:**

a. In the first phase, which is the training phase, the naive Bayes classifier reads in the training data along with the training labels and learns the parameters used by the classifier. **(20 points)**

b. In the testing phase, the trained naive Bayes classifier classifies the data in the testing data file. You will need to convert the articles in the testing data into a feature vector, just like in the training data where a 1 in the ith slot indicates the presence of the ith word in the vocabulary while a 0 indicates the absence. If you encounter a word in the testing data that is not present in your vocabulary, ignore that word. Note that the feature vector is only of size M because the class labels are not part of the testing data. **(20 points)**

c. Output the accuracy of the naive Bayes classifier by comparing the predicted class label of each message in the testing data to the actual class label. The accuracy is the number of correct predictions divided by the total number of predictions. **(10 points)**

RESULTS:

♦ For testing data, print out each article’s actual class label and classified class label as follows: **(5 points)**

```
-------------------------------------
Actual class: ARXIV
Classified class: JDM
-------------------------------------
Actual class: PLOS
Classified class: PLOS
-------------------------------------
```

♦ Finally print out accuracy result for each class.