NLP assignment2 Text Classification

姓名：王涵 学号：2015211984

```
Keywords:
Text Classification, Naive Bayes, K-Nearest Neighbours, Supported Vector Machine
```

Problem 1

One of the main ML problems is text classification, which is used, for example, to detect spam, define the topic of a news article, or choose the correct mining of a multi-valued word.

This data set contains 1000 text articles posted to each of 20 online newgroups, for a total of 20,000 articles. For documentation and download, see http://www-2.cs.cmu.edu/afs/cs/project/theo-11/www/naive-bayes.html.

To solve this problem, there are two major method.
(1) Count the word to compute the posterior probablities of word of text. Then apply Naive Bayes Algorithm to the model.
(2) Vectorize the document using tf-idf representation. Then apply vector-supported machine learning algorithm to do classification, such algorithms can be SVM, KNN and Decision Tree,etc.

Usage

Software Requirement

```
python 2.7.12
sklearn=0.19
numpy=1.12
nltk=3.2
```

Source Code

Jupyter Notebook

text-classification.py

Dataset

This data set contains 1000 text articles posted to each of 20 online newgroups, for a total of 20,000 articles.

The content of the dataset is almost all mail-styled with header of enormous length.

Preprocessing

In order to eliminating the disturbing factors of the text content and further reduce the length of the word bank only to preserve the useful words. The preprossing is carried out by the following steps:

(1) Remove the useless header.

The information in the header is uninformative.

(2) Remove the word in e-mail style or contains more than one digit.

Words in both styles can be considered with no information.

(3) Remove the stopwords

(4) Word Stemming.

After applying a NLTK supported Word Net Stemming API, the length of word bank is reduced to about 25,000 comparing to a 90,000 before stemming and other preprecessing steps.

(5) Document Vectorization

Using tf-idf with Laplace Smoothing to find the vector representation of a document.

Algorithms and Models

(1) Naive Bayes

First, apply **maximum likelihood estimation** to compute the posterior probablities of words in text classes. Then, compute the probablity to get the maximum probablity and its corresponding label.

(2) K-Nearest Neighbors

Training step:

Record all X and Y on the training set.

Test Step:
Find the k nearest neighbor to represent the vector which determines the label of the testing sample

(3) Support Vector Machine

Find a hyperplane to seperate the vector space into two classes. When applying to multiclass classification problem. It is considered to separate several times.

Perfomance

	# of word	Naive Bayes Acc.	KNN Acc.(k=1)	KNN Acc.(k=3)	SVM Acc.
1	25355	75.250 %	67.240 %	63.371 %	78.220 %
2	25878	74.425 %	67.005 %	63.277 %	77.520 %
3	25734	75.200 %	67.224 %	62.531 %	81.250 %

Analysis

From the charts showed above, the Naive Bayes models and SVM model perform almost equally on this dataset. For 75% accuracy almost reached the theoretical limit of Naive Bayes.

While training on the SVM model, input shape of (19997,25000) is almost infinite time to training to classify between 20 classes.

However,KNN doing poorly whether with k=1 and k=3. It may be the cause of wrongly selected k. Due to the it gets more time-consuming to test while k gets larger, my decision is that not to experiment over and over again.