Spam Email Classifier

SUBMITTED TO:

SUBMITTED BY:

MR. NITIN

YASH RAJ
101511069

NLTK used for

- Tokenizing words
- Lemmatizing words
- Removing stop words from the training data
- Using Naïve Bayes Classifier for classification purposes

Code:

```
import nltk
import os
import random
from nltk import word_tokenize, WordNetLemmatizer
from nltk.corpus import stopwords
from nltk import NaiveBayesClassifier, classify

stoplist = stopwords.words('english')

def get_data(folder):
    a = []
    files = os.listdir(folder)
    for file in files:
        f = open(folder + file, 'r', encoding = "ISO-8859-1")  ## cause there are some latin
words also
        a.append(f.read())
    f.close()
    return a
```

Necessary imports

Creating list of stop words from English language

Reading all the files

`

takes sentence as parameter and lemmatize it word by word

```
def preprocess(sentence):
lemmatizer = WordNetLemmatizer()
   return [lemmatizer.lemmatize(word.lower()) for word in word_tokenize(sentence)]

def getfeatures(text):   #creating feature list
    return {word: True for word in preprocess(text) if not word in stoplist}

def print_accuracy(train_set, test_set, classifier): #finding accuracy of the training and
testing data
   print ('Accuracy on the training set = ' + str(classify.accuracy(classifier, train_set)))
   print ('Accuracy of the test set = ' + str(classify.accuracy(classifier, test_set)))
   classifier.show_most_informative_features(10)
```

```
def train(features, samples_proportion):
    train_size = int(len(features) * samples_proportion)

    train_set, test_set = features[:train_size], features[train_size:]
    print ('Training set size = ' + str(len(train_set)) + ' emails')
    print ('Test set size = ' + str(len(test_set)) + ' emails')

    classifier = NaiveBayesClassifier.train(train_set)
    return train_set, test_set, classifier
```

#finds index of data at 80%

#defining training and #testing data

#training using naïve Bayes #classifier

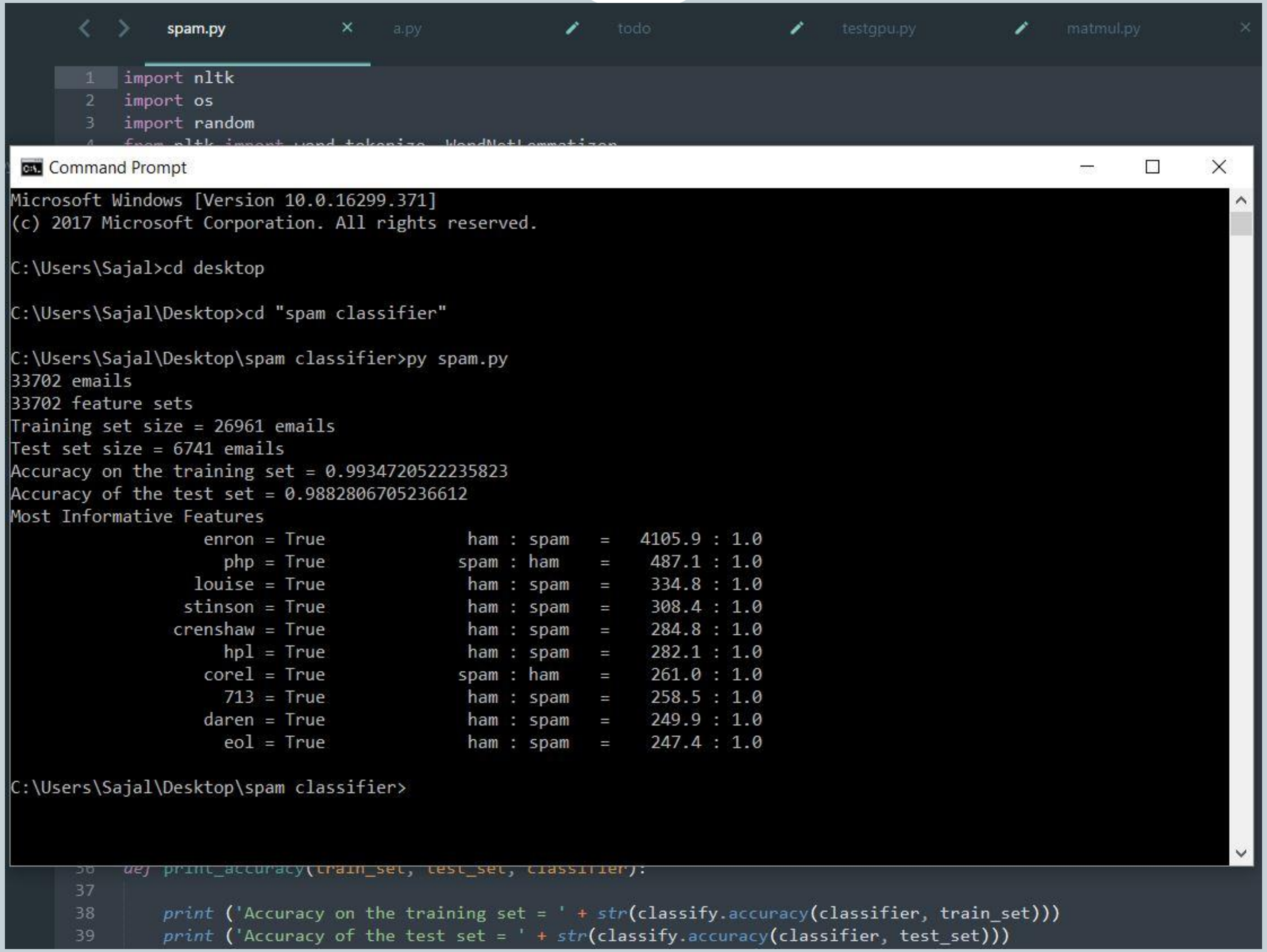
```
if __name__ == "__main__":
    spam = get_data('enron1/spam/')        #reads spam emails
    ham  = get_data('enron1/ham/')                          #reads ham emails

    allemails  = [(email, 'spam') for email in spam]#creating a list of tuples having email
    allemails += [(email, 'ham') for email in ham]      with  its label
    random.shuffle(allemails)                           #shuffling the list to randomly distribute
                                                               training and testing data
    print (str(len(allemails)) + ' emails')

    allfeatures = [(getfeatures(email), label) for (email, label) in allemails]   #learning features
    print (str(len(allfeatures)) + ' feature sets')                                   from the
data


    train_set, test_set, classifier = train(allfeatures, 0.8)    #defining size of training ,testing
data
```

Result



```
import nltk
import os
import random
from nltk import word_tokenize, WordNetLemmatizer
```

```
Microsoft Windows [Version 10.0.16299.371]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Sajal>cd desktop

C:\Users\Sajal\Desktop>cd "spam classifier"

C:\Users\Sajal\Desktop\spam classifier>py spam.py
33702 emails
33702 feature sets
Training set size = 26961 emails
Test set size = 6741 emails
Accuracy on the training set = 0.9934720522235823
Accuracy of the test set = 0.9882806705236612
Most Informative Features
                   enron = True              ham : spam   =   4105.9 : 1.0
                     php = True             spam : ham    =    487.1 : 1.0
                  louise = True              ham : spam   =    334.8 : 1.0
                 stinson = True              ham : spam   =    308.4 : 1.0
                crenshaw = True              ham : spam   =    284.8 : 1.0
                     hpl = True              ham : spam   =    282.1 : 1.0
                   corel = True             spam : ham    =    261.0 : 1.0
                     713 = True              ham : spam   =    258.5 : 1.0
                   daren = True              ham : spam   =    249.9 : 1.0
                     eol = True              ham : spam   =    247.4 : 1.0

C:\Users\Sajal\Desktop\spam classifier>
```

```
def print_accuracy(train_set, test_set, classifier):

    print ('Accuracy on the training set = ' + str(classify.accuracy(classifier, train_set)))
    print ('Accuracy of the test set = ' + str(classify.accuracy(classifier, test_set)))
```