**Name: Sandhya Sirisha Bhamidipati**

**UTA ID: 1001663914**

**Problem statement:**

“Spam” is a term used for unsolicited bulk email messages. Spam may encompass everything from ads for products and services, drugs, pornographic content, money scams, stock market pump-and-dump schemes, malware, phishing and everything in between. Solution to this is to filter the spam mails so that the recipient has no need to pay the cost for these phishing mails.

**Approach:**

**Spam Filtering based on Naive Bayes Classification:**

Implementation of a simple machine learning algorithm to classify emails (Spam or Not Spam aka Ham).

We are going to implement a Spam filtering classifier based on Multinomial Naive Bayes Algorithm without the use of sklearn

We are going to solve this spam filtering problem as follows:

\* Prepare the data

\* Build the training data

\* Train the classifier

\* Test the classifier for its accuracy

\* Classify a new email from test data with our trained classifier

**Prepare the data:**

The train and test dataset have been downloaded from the publicly available Lingspam dataset (http://www.aueb.gr/users/ion/data/lingspam\_public.tar.gz).

We will use the `bare > part1` from the downloaded dataset as the training data and `bare > part2` as the test data.

So, in brief the process is divided into the following steps:

**On training data:**

a. Build the dictionary

b. Build training features and labels

c. Train the classifier

**On testing data:**

d. Build the test features and labels

e. Calculate accuracy of the trained classifier

Initially, the dataset is prepared by splitting the sentences into individual words and deleting all the punctuations, delimiters etc. Later, the duplicates of the words, if any, are removed.

Later, we should train the prepared data. We have a class called Multinomial Naive Bayes where the required calculations for this algorithm are required and converting them into logarithmic values.

Later, we train our dataset on that classifier and calculate the accuracy and then we do the same for test data and calculate the accuracy.

Here, we use Laplace smoothing, which is a technique for smoothing categorical data. A small-sample correction, or pseudo-count, will be incorporated in every probability estimate. Consequently, no probability will be zero. This is a way of regularizing Naive Bayes.

**Directions to run the spamfilter.py:**

Make sure all the required files are in the same directory. Open command prompt and then change to the directory where these files are present. Now, type python spamfilter.py to run the file. Then the output looks as below:

*1. Building the dictionary*

*2. Building the training features and labels*

*3. Training the classifier*

*Train Accuracy: 99.6540%*

*4. Building the test features and labels*

*5. Calculating the accuracy of the trained classifier over the test emails*

*Test Accuracy: 97.7273%*

**References:**

1. numpy (http://www.numpy.org/), a powerful N-dimensional array object

2. Multinomial Naive Bayes Algorithm (https://monkeylearn.com/blog/practical-explanation-naive-bayes-classifier/)

3. https://towardsdatascience.com/spam-classifier-in-python-from-scratch-27a98ddd8e73

4. <https://medium.com/syncedreview/applying-multinomial-naive-bayes-to-nlp-problems-a-practical-explanation-4f5271768ebf>

5. <https://www.pythonforengineers.com/build-a-spam-filter/>

**Experiences:**

Really enjoyed doing this project for the reason that I had learnt a lot of new techniques. Also, encountered some difficulties because I did not use the inbuilt function “sklearn” for this project. So, I had to create a separate class which does the task for us during which I was stuck for a while but later I resumed once I was able to debug and fix my errors.