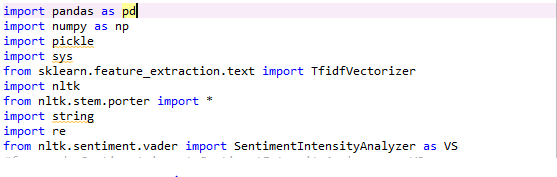
**IMPLEMENTATION DETAILS**

The project is divided into the following sub-divisions:

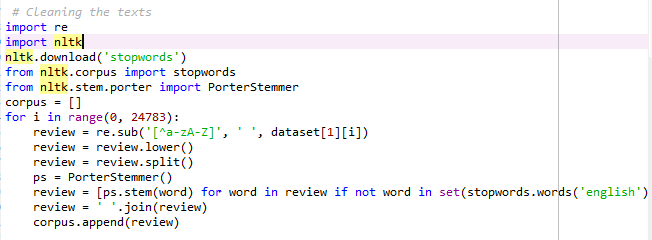
**1. Importing required libraries:**

* Libraries used here are pandas, numpy, nltk, string etc..



**2. Pre-processing the input text:**

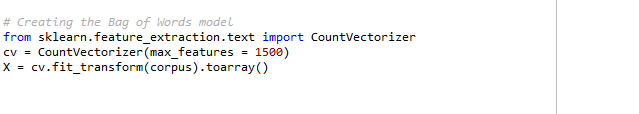
* In Data Pre-processing the main step is to removal of stop words and punctuation marks
* A stop word is a commonly used word (such as “the”, “a”, “an”, “in”) that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query.



**3. Vectorization:**

* We will be creating vectors that have a dimensionality equal to the size of our vocabulary, and if the text data features that vocab word, we will put a one in that dimension (column of bag of words matrix)
* Every time we encounter that word again, we will increase the count, leaving 0s everywhere we did not find the word even once.

**4.** **Build a Model:**



* The algorithms that we have implemented are as follows:

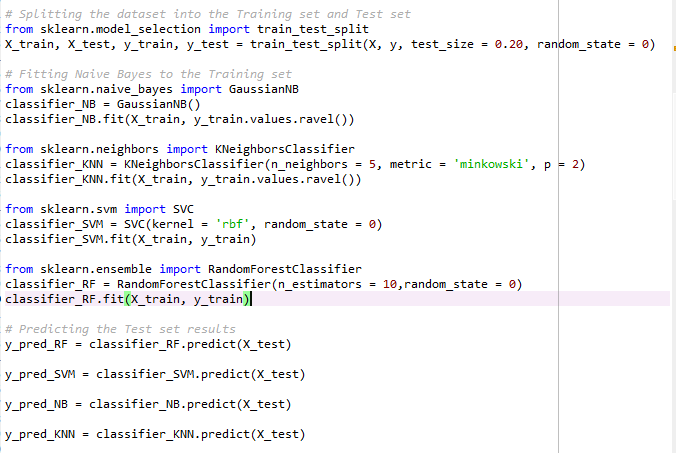
1.KNN

2.SVM

3.Naive Bayes

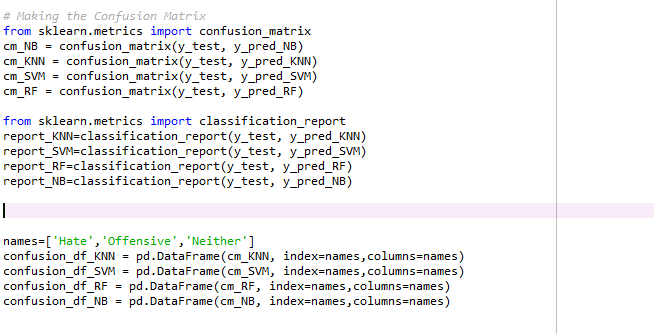
4.Random Forest

* We divide the data set into training set and test set. The above algorithms are trained with the training set.



**5. Evaluate a model:**

* After training all the models the performance of each algorithm is evaluated using the test set.
* The test set is fed into the models and the output is predicted.
* Confusion matrices are drawn to evaluate the efficiencies of all the models.
* The model with highest efficiency is preferred over others.



**6. Build a frontend to display the reports**

* tkinter is a GUI library in python which is used to make user interfaces
* we create a blank window by default
* it is used to display static text which we pass as argument to the attribute
* grid function is used to give layout of various elements in the window
* config function is used to text formatting and styling.
* button is used to provide interactivity upon user actions in the window.

