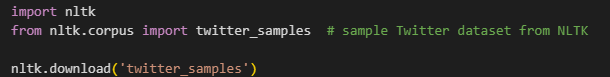
--Project A--

The problem is to determine whether a given Tweet has a positive or negative sentiment. First, you will   
need  to  perform  data  cleaning  by  removing  punctuation,  removing  stop  words  and  performing   
stemming. Next, you will split the dataset between training and testing. Finally, you will have to classify   
correctly the tweets of the testing set in either positive or negative.

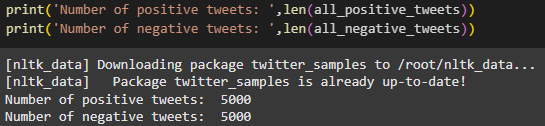
Firstly we need to download the data sets:



Then we select the set of positive and negative tweets:



We can also show the length of the both tweets:

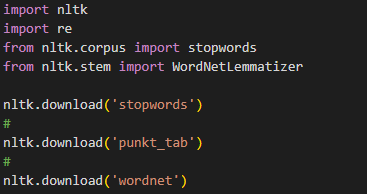


The next step is to perform Data Cleaning:

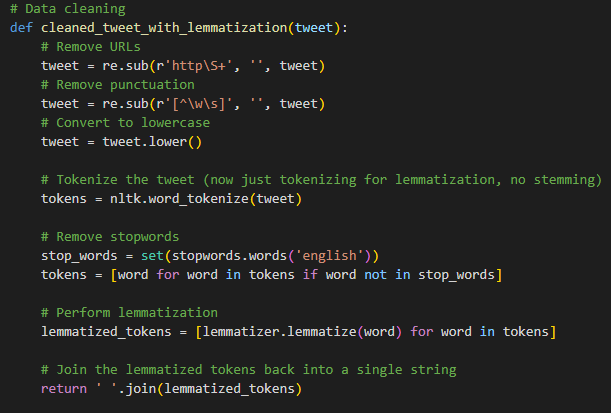
* Data Cleaning

Data Cleaning contains the process of removing stop words(common words that are excluded while performing data cleaning), We also remove punctuation marks, url, emojis. An important process of Data Cleaning is also Stemming or Lemmatization. In this project I chose to perform Lemmatization because its more practical and after performing it the word still has a meaning, not as in Stemming where run🡺 ru or studies 🡺 studi.

In order to peform Data Cleaning, python has a variety of libraries which can help us to just call them:



The next step is to make a function that can process lemmatization but also remove stopwords, punctuation marks, etc.



Together we are gonna explain what every line of the code does:

First We make a function called cleaned\_tweet\_with\_lemmatization() that accept a parameter. The function contains:

Removing the URLs.

Removing punctuation.

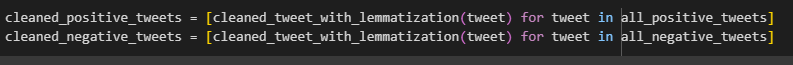
Converting the string into lowercase(so it can be the same)

Performing tokenization for lemmatization.

Removing stopwords.

And in the end of the code we have to joing or append every token back together into a string, so it can be in the line.

The next step is to apply the function into the data sets:



In order to turn the data sets into numerical so that the machine can read them, We have to combine them:



We also have to make a label for them(0 for negatives and 1 for positives):



After the data sets are cleaned and combined, We can move on to the next part which is to convert the data sets into the TF-IDF vectors:

* TF-IDF

In order to peform TF-IDF, We have to import the libraries about the conversion:



Now we can apply the TF-IDF:



* Data splitting

After we converted the data to TF-IDF, We can now split the data into training set(80%) and testing set(20%). Also sklearn can help us peform the data splitting:



* Model Training

In this project, I’ve chosen the SVM(Support Vector Machines) algorithm, because it is the best usable algorithm for sentiment analysis.

First we need to import the algorithm:



We initialize the SVM model:



The next step is to train the SVM model:

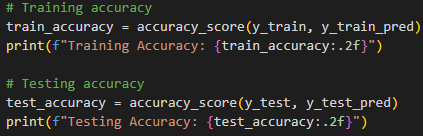


The next step is to predict on the training set and the testing set:



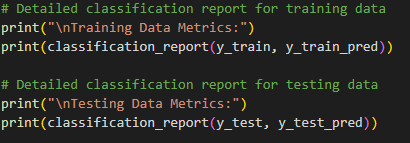
* Classification report

After we are done with model training, we can see the results of the prediction of the training accuracy and testing accuracy:



The “2.f” is used to show just the two digits after floating point.

We also show the detailed classification report for the training and testing data:



The output of the following code is:

