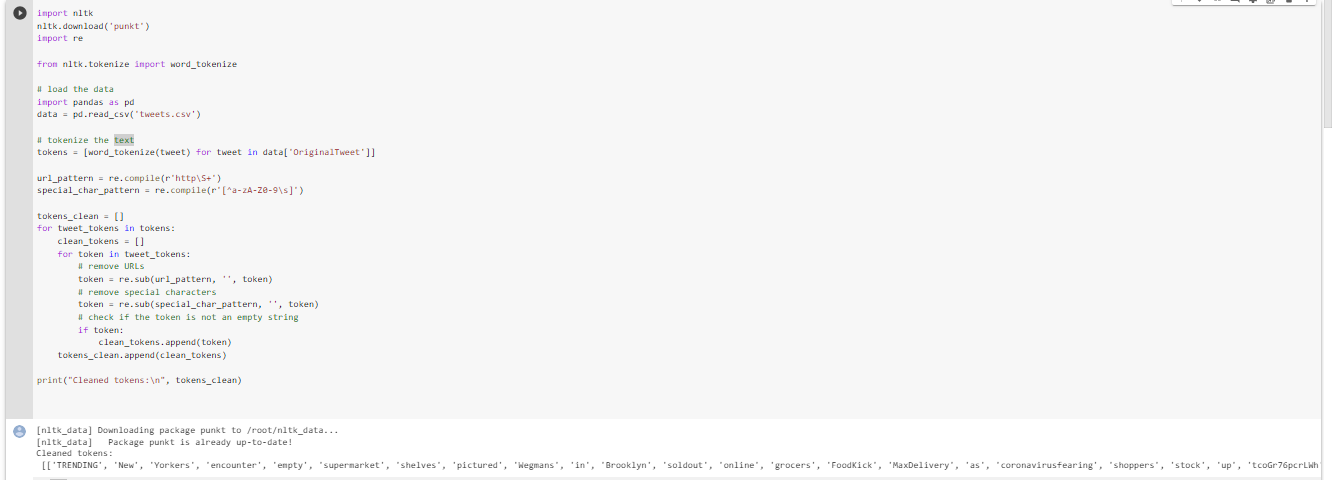
**PDS ASSIGNMENT 3**

**BHASHITHA JAGARLAMUDI**

1) (20 points) The data file contains tweets that have been pulled from Twitter. In this dataset  
use the text data in the “OriginalTweet” column and perform the following:  
a) Convert the text corpus into tokens.  
b) Perform stop word removal.  
c) Count Word frequencies  
d) Create word clouds

a) Convert the text corpus into tokens:

In order to tokenize and sanitize text data from a csv file containing tweets, this code imports the required libraries. The text data is tokenized using the Natural Language Toolkit (nltk) in the code, and URLs and other special characters are eliminated by regular expressions (re). After being tokenized with nltk and imported into a pandas DataFrame, the text data is saved in a list. The for loop cycles through the list of tokens and applies regular expressions on each token's cleaning. The code cleans up text data and gets it ready for further analysis using regular expressions and nltk



b) Perform stop word removal.

The'stop\_words' collection has been loaded with the English language's list of stopwords. The initial state of the 'filtered\_tokens' list is an empty list. The for loop repeatedly goes through the list of cleaned tokens and uses a list comprehension to generate a fresh list of tokens without stopwords. The 'filtered\_tokens' list is then appended with the filtered tokens that were obtained.



c) Count Word frequencies

From the collections library, Counter is imported and Initializatied the 'word\_freq' counter results in an empty counter object. The for loop iterates through the list of tokens that have been filtered, updating the 'word\_freq' counter object with the frequency of each word in the tweet. To display the word frequencies as a dictionary, the 'word\_freq' counter object is finally transformed into a dictionary and printed once more. For the purpose of text analysis and visualization, I used the collections library to count the word frequencies in a list of filtered tokens.



d) Create word clouds

Using the WordCloud library and the 'word\_freq' dictionary produced in the previous phase, I generated a word cloud display. The 'generate\_from\_frequencies' method is used to build a WordCloud object with the provided width, height, and background color. This object is then generated from the 'word\_freq' dictionary. The generated word cloud is plotted using the pyplot 'imshow' function with the 'axis' off to eliminate any axis labels or ticks, and the plot is then presented using 'plt.show()'. To visualize the frequency of terms in the filtered token data, this code generates a word cloud visualization in its entirety.

