**Report**

The code imports all the necessary libraries such as pandas, regular expression (re), natural language toolkit (nltk), stopwords, and collections. It then reads a CSV file called "Corona\_NLP\_test.csv" and saves it into a pandas dataframe called "Tweets".

The text data in the "OriginalTweet" column is preprocessed by removing URLs, mentions, hashtags, and punctuations using regular expressions. It is then tokenized using nltk's word\_tokenize() function, and stop words are removed using the stopwords.words() function. The resulting cleaned tweets are added to a new column in the "Tweets" dataframe called "clean\_text". The cleaned dataset is then saved as a new CSV file called "cleaned\_tweets.csv".

The code then tokenizes the cleaned tweets and removes stop words again, resulting in a list of filtered tokens. Using the Counter class from collections, it counts the frequencies of each token and saves the counts in a dictionary called "word\_freq".

Finally, the code uses the wordcloud library to create a word cloud from the word frequencies dictionary, with certain formatting options such as white background color and minimum font size. The resulting word cloud is displayed using matplotlib's imshow() function.

Overall, the code performs data preprocessing and cleaning, counts word frequencies, and creates a word cloud visualization.

**a) Convert the text corpus into tokens:**

In this step, I have converted the text corpus into tokens using the NLTK library's word\_tokenize() function.

**b) Perform stop word removal:**

I have performed stop word removal using the NLTK library's set of stop words in English. Then i removed all the stop words from the text corpus.

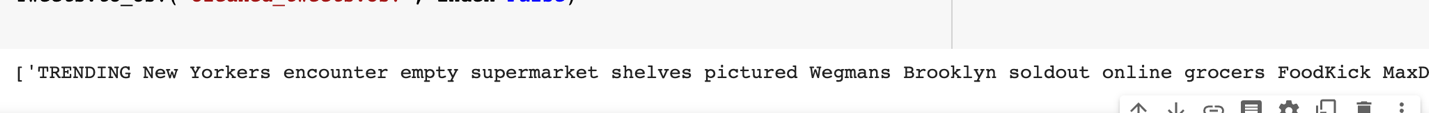
**c) Count Word frequencies:**

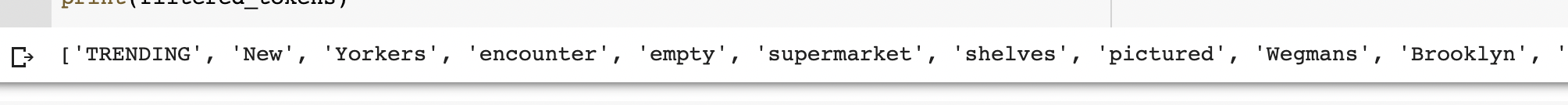
I have counted the frequency of each word using the Counter function from the collection’s library. This provides me with the number of occurrences of each word in the text corpus.

**d) Create word clouds:**

I have created a word cloud using the WordCloud library in Python. A word cloud is a visualization of word frequency in a text corpus. In the word cloud, the size of each word represents its frequency in the text corpus. Words that occur more frequently are displayed larger and more prominently. I have created a word cloud using the word frequencies obtained in step c) and plotted it using the matplotlib library.

**Results**:







A picture containing text

Description automatically generated