import numpy as np

print('numpy: {}'.format(np.\_\_version\_\_)) # Printing the version of Numpy

import pandas as pd

print('pandas: {}'.format(pd.\_\_version\_\_)) # Printing the version of pandas

import re

import matplotlib.pyplot as plt

%matplotlib inline

import re

import nltk

nltk.download('wordnet')

nltk.download('stopwords')

from nltk.corpus import stopwords

from termcolor import colored

from nltk.stem import PorterStemmer

from nltk.stem import WordNetLemmatizer

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_--

Step-2

# Read 01\_first\_round\_cleaned\_tweets\_df.csv file

# Check missing value distribution

# Find Class distribution

# Processing tweet text data

Removing user handles starting with @ - str.replace("@[\w]\*","")

Removing numbers and special characters - str.replace("[^a-zA-Z' ]",""

Removing urls - ((www\.[^\s]+)|(https?://[^\s]+))

Removing single characters - r"(^| ).( |$)"), " "

Tokenizing - Check notebook

Removing stopwords - such as a and the,

Expanding not words ( Expanding "n't" to not)

Stemming the words

**Stemming - two or more words have a common root. For example, the three words – agreed, agreeing and agreeable have the same root word agree. A search involving any of these words should treat them as the same word which is the root word.**

**Combining words back to tweets.**

**Use - from nltk.stem import PorterStemmer**

<https://www.guru99.com/stemming-lemmatization-python-nltk.html>

Lemmatizing the words

**Lemmatization is similar to stemming but it brings context to the words.So it goes a steps further by linking words with similar meaning to one word. For example if a paragraph has words like cars, trains and automobile, then it will link all of them to automobile**

**Use - from nltk.stem import WordNetLemmatizer**

<https://www.guru99.com/stemming-lemmatization-python-nltk.html>

Combining words back to tweets

# Check missing value distribution again

# Arrange the columns

['tweet\_id', 'SourceDataBase', 'OS', 'Tweet-Class\_category-Code',

'Tweet\_source', 'Tweeted-By', 'retweet\_count', 'Tweet', 'Clean\_tweet', 'Country',

'class']

# Check again Positive , Negative and Neutral tweets Wordcloud with “cleantweet”

# Create a df with new arrangement and write to csv ( like “02\_2nd\_round\_cleaned\_tweets\_df.csv”)

#### Tokenenization

tweet = "I finally got through on the phone after a 2 hour wait time"

x = tweet.split()

print(x)

End of Step-2