**Steps for Text Cleaning**

1. In this project, we are going to clean the text using NLTK library. You are required to download it first.

! pip install nltk

1. Open the csv file

import pandas as pd

data = pd.read\_csv("tweets\_Ocean.csv")

1. Tweets contains URL, special characters. It is time to remove those because you will not get good results without cleaning the tweets. Here we will use the concept of regular expression . So, you are required to import re.

**####Remove URL from Tweets**

data['Tweets without URL'] = data['Tweets'].apply(lambda x : re.sub(r'https?:\/\/\S\*',"",x) )

**# removing special characters and number**s

data['Tweets without special characters'] = data['Tweets without URL'].apply(lambda x : re.sub("[^(a-z)^(A-Z)^(0-9)^%\s]","",x) )

**# remove \n(new line character)**

data['Tweets without special characters'] = data['Tweets without special characters'].apply (lambda x : re.sub("\n", " ",x) )

1. Convert the tweets to lower case(optional, depends upon the requirement)

**###Convert the tweets into lower case**

data['Tweets without special characters']=data['Tweets without special characters'].str.lower()

1. **Stopwords** : Stop words are basically a set of commonly used words in any language, not just English like it, in, how, we etc. It is worth to remove these words from tweets or from text because it will unnecessarily take storage and computing power despite of no use.

**####Remove stopwords**

stopword = nltk.corpus.stopwords.words('english')

data['Tweets without stop words'] = data['Tweets without special characters'].apply(lambda x :" ".join(word for word in x.split() if word not in stopword ))

1. **Tokenization**: It is important to convert text into tokens before transforming it into vectors. Vectors are required to train ML/DL models. Divide tweets into tokens

data['Tweet\_tokenized'] = data['Tweets without stop words'].apply(lambda x:re.split('\W+', x))

1. **Stemming**(optional, depends upon the requirement): Stemming is the process of reducing a word to its word stem. For **example**, the **stem** of the words studied🡪studi, studying🡪study

ps = nltk.PorterStemmer()

def stemming(text):

text = [ps.stem(word) for word in text]

return text

1. **Lemmatization**(optional, depends upon the requirement): Lemmatization considers the context and converts the word to its meaningful base form. E.g. Studies🡪study, studying🡪study

nltk.download('wordnet')

wn = nltk.WordNetLemmatizer()

def lemmatizer(text):

text = [wn.lemmatize(word) for word in text]

return text

1. Since the dataset is related to the tweets posted by the users on the World Ocean Day which was on June 8, 2021, We are interested to see the most popular words in the tweets by creating cloud of word. To create that, you are required to download wordcloud

! pip install wordcloud

###Display cloud

from wordcloud import WordCloud, STOPWORDS , ImageColorGenerator

tweet\_All = " ".join(review for review in data['Tweets without special characters'])

fig, ax = plt.subplots(figsize = (30,30))

# Create and generate a word cloud image:

wordcloud\_ALL = WordCloud(max\_font\_size=50, max\_words=100, background\_color="white").generate(tweet\_All)

ax.imshow(wordcloud\_ALL, interpolation='bilinear')

ax.set\_title('All Tweets', fontsize=30)

ax.axis('off')