In [1]:
```
pip install nltk
```

```
Collecting nltk
  Downloading nltk-3.8.1-py3-none-any.whl.metadata (2.8 kB)
Requirement already satisfied: click in c:\users\rohan\appdata\local\programs\python
\python310\lib\site-packages (from nltk) (8.1.7)
Requirement already satisfied: joblib in c:\users\rohan\appdata\local\programs\pytho
n\python310\lib\site-packages (from nltk) (1.3.2)
Collecting regex>=2021.8.3 (from nltk)
  Downloading regex-2023.12.25-cp310-cp310-win_amd64.whl.metadata (41 kB)
     ---------------------------------------- 0.0/42.0 kB ? eta -:--:--
     ---------------------------------------- 42.0/42.0 kB 1.0 MB/s eta 0:00:00
Requirement already satisfied: tqdm in c:\users\rohan\appdata\local\programs\python
\python310\lib\site-packages (from nltk) (4.66.2)
Requirement already satisfied: colorama in c:\users\rohan\appdata\local\programs\pyt
hon\python310\lib\site-packages (from click->nltk) (0.4.6)
Downloading nltk-3.8.1-py3-none-any.whl (1.5 MB)
   ---------------------------------------- 0.0/1.5 MB ? eta -:--:--
   -- ------------------------------------- 0.1/1.5 MB 1.7 MB/s eta 0:00:01
   ---- ----------------------------------- 0.2/1.5 MB 1.9 MB/s eta 0:00:01
   ----------- ---------------------------- 0.5/1.5 MB 3.1 MB/s eta 0:00:01
   ---------------------- ----------------- 0.8/1.5 MB 4.5 MB/s eta 0:00:01
   ------------------------------------- -- 1.4/1.5 MB 6.4 MB/s eta 0:00:01
   ------------------------------------- -- 1.4/1.5 MB 6.4 MB/s eta 0:00:01
   ------------------------------------- -- 1.4/1.5 MB 6.4 MB/s eta 0:00:01
   ------------------------------------- -- 1.4/1.5 MB 6.4 MB/s eta 0:00:01
   ------------------------------------- -- 1.4/1.5 MB 6.4 MB/s eta 0:00:01
   ------------------------------------- -- 1.4/1.5 MB 6.4 MB/s eta 0:00:01
   ------------------------------------- -- 1.4/1.5 MB 6.4 MB/s eta 0:00:01
   ------------------------------------- -- 1.4/1.5 MB 6.4 MB/s eta 0:00:01
   ------------------------------------- -- 1.4/1.5 MB 6.4 MB/s eta 0:00:01
   ---------------------------------------  1.5/1.5 MB 2.3 MB/s eta 0:00:01
   ---------------------------------------  1.5/1.5 MB 2.3 MB/s eta 0:00:01
   ---------------------------------------- 1.5/1.5 MB 2.0 MB/s eta 0:00:00
Downloading regex-2023.12.25-cp310-cp310-win_amd64.whl (269 kB)
   ---------------------------------------- 0.0/269.5 kB ? eta -:--:--
   ---------------- ----------------------- 112.6/269.5 kB ? eta -:--:--
   ---------------------------------------- 269.5/269.5 kB 3.3 MB/s eta 0:00:00
Installing collected packages: regex, nltk
Successfully installed nltk-3.8.1 regex-2023.12.25
Note: you may need to restart the kernel to use updated packages.
```

In [2]:
```python
import nltk as nltk
nltk.download("punkt")
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('averaged_perceptron_tagger')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Rohan\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping tokenizers\punkt.zip.
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Rohan\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\stopwords.zip.
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\Rohan\AppData\Roaming\nltk_data...
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\Rohan\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping taggers\averaged_perceptron_tagger.zip.
```

Out[2]: True

In [4]:
```
text= "Tokenization is the first step in text analytics. The process of breaking do
```

In [5]:
```
from nltk.tokenize import sent_tokenize
tokenized_text= sent_tokenize(text)
print(tokenized_text)
```

```
['Tokenization is the first step in text analytics.', 'The process of breaking down
a text paragraph into smaller chunks such as words or sentences is called Tokenizati
on.']
```

In [6]:
```
from nltk.tokenize import word_tokenize
tokenized_word=word_tokenize(text)
print(tokenized_word)
```

```
['Tokenization', 'is', 'the', 'first', 'step', 'in', 'text', 'analytics', '.', 'Th
e', 'process', 'of', 'breaking', 'down', 'a', 'text', 'paragraph', 'into', 'smalle
r', 'chunks', 'such', 'as', 'words', 'or', 'sentences', 'is', 'called', 'Tokenizatio
n', '.']
```

In [7]:
```
import regex as re
from nltk.corpus import stopwords
stop_words=set(stopwords.words("english"))
print(stop_words)
text= "How to remove stop words with NLTK library in Python?"
text= re.sub('[^a-zA-Z]', ' ',text)
tokens = word_tokenize(text.lower())
filtered_text=[]
for w in tokens:
   if w not in stop_words:
      filtered_text.append(w)
print("Tokenized Sentence:",tokens)
print("Filterd Sentence:",filtered_text)
```

```
{'her', 'does', 'did', 'same', 'each', 're', 'ourselves', 'mustn', 'above', 'yoursel
f', "didn't", "wasn't", 'will', 'doesn', "mustn't", "needn't", 'who', "doesn't", 'i
t', 'an', 'be', 'other', 'haven', 'themselves', "you'd", 'we', 'y', 'll', 'she', 'hi
s', 'had', 'hers', 'don', 'further', 'am', "won't", 'there', 'below', 'ours', "you'v
e", 'or', 'because', 'up', 'weren', 'him', 'few', 'me', 'our', 'by', 'i', 'most', 'c
an', 'that', 'herself', 'a', "hasn't", 'here', 'are', 'been', 'both', 'my', "it's",
'than', "haven't", 'needn', 'your', 'hasn', 'then', 'itself', 'couldn', 'doing', 's
o', 'before', 'shouldn', 'with', 'against', 'only', 'no', 'theirs', 'yours', 'm',
't', 'into', "shan't", 'about', 'such', 'some', 'wouldn', 'not', 'mightn', "she's",
'isn', 'to', 'any', 'these', 'now', 'himself', 'whom', "mightn't", 've', 'they', "do
n't", 'until', 'from', 'their', "you'll", 'in', 'own', 'wasn', "wouldn't", 'of',
'o', 'on', 'being', 'if', 'during', 'them', 'd', 'was', 'those', 'have', 'you', 'wer
e', "aren't", 'its', 'ain', 'for', 'and', 'is', 'over', "should've", 'which', "tha
t'll", 'once', 'where', 'yourselves', 'should', 'he', 'off', 'nor', "weren't", 'afte
r', "you're", "hadn't", 'the', 'through', "shouldn't", 'this', 'as', 'again', 'unde
r', 'what', "isn't", 'how', 's', "couldn't", 'do', 'won', 'myself', 'having', 'are
n', 'all', 'too', 'didn', 'when', 'between', 'hadn', 'ma', 'shan', 'very', 'while',
'more', 'but', 'why', 'at', 'down', 'out', 'just', 'has'}
Tokenized Sentence: ['how', 'to', 'remove', 'stop', 'words', 'with', 'nltk', 'librar
y', 'in', 'python']
Filterd Sentence: ['remove', 'stop', 'words', 'nltk', 'library', 'python']
```

In [8]:
```
pip install regex
```

```
Requirement already satisfied: regex in c:\users\rohan\appdata\local\programs\python
\python310\lib\site-packages (2023.12.25)
Note: you may need to restart the kernel to use updated packages.
```

In [9]:
```
from nltk.stem import PorterStemmer
e_words= ["wait", "waiting", "waited","waits"]
ps =PorterStemmer()
for w in e_words:
   rootWord=ps.stem(w)
print(rootWord)
```

```
wait
```

In [10]:
```
from nltk.stem import WordNetLemmatizer
wordnet_lemmatizer =WordNetLemmatizer()
text = "studies studying cries cry"
tokenization =nltk.word_tokenize(text)
for w in tokenization:
   print("Lemma for {} is{}".format(w,
wordnet_lemmatizer.lemmatize(w)))
```

```
Lemma for studies isstudy
Lemma for studying isstudying
Lemma for cries iscry
Lemma for cry iscry
```

In [11]:
```
import nltk
from nltk.tokenize import word_tokenize
data="The pink sweater fit her perfectly"
words=word_tokenize(data)
for word in words:
    print(nltk.pos_tag([word]))
```

```
[('The', 'DT')]
[('pink', 'NN')]
[('sweater', 'NN')]
[('fit', 'NN')]
[('her', 'PRP$')]
[('perfectly', 'RB')]
```

In [12]:
```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
```

```
C:\Users\Rohan\AppData\Local\Temp\ipykernel_18216\3080576706.py:1: DeprecationWarnin
g:
Pyarrow will become a required dependency of pandas in the next major release of pan
das (pandas 3.0),
(to allow more performant data types, such as the Arrow string type, and better inte
roperability with other libraries)
but was not found to be installed on your system.
If this would cause problems for you,
please provide us feedback at https://github.com/pandas-dev/pandas/issues/54466

  import pandas as pd
```

In [13]:
```
documentA = 'Jupiter is the largest Planet'
documentB = 'Mars is the fourth planet from the Sun'
```

In [14]:
```
bagOfWordsA = documentA.split(' ')
bagOfWordsB = documentB.split(' ')
```

In [15]:
```
uniqueWords =set (bagOfWordsA).union(set(bagOfWordsB))
```

In [16]:
```
numOfWordsA = dict.fromkeys(uniqueWords, 0)
for word in bagOfWordsA:
    numOfWordsA[word] += 1
    numOfWordsB = dict.fromkeys(uniqueWords,0)
    for word in bagOfWordsB:
        numOfWordsB[word] += 1
```

In [17]:
```
def computeTF(wordDict, bagOfWords):
   tfDict = {}
   bagOfWordsCount =len(bagOfWords)
   for word, count in wordDict.items():
      tfDict[word] = count / float (bagOfWordsCount)
   return tfDict
tfA = computeTF(numOfWordsA,bagOfWordsA)
tfB =computeTF(numOfWordsB, bagOfWordsB)
tfA
tfB
```

Out[17]:
```
{'Sun': 0.125,
 'the': 0.25,
 'is': 0.125,
 'fourth': 0.125,
 'largest': 0.0,
 'Planet': 0.0,
 'from': 0.125,
 'Mars': 0.125,
 'planet': 0.125,
 'Jupiter': 0.0}
```

In [18]:
```
def computeIDF(documents):
     import math
     N = len(documents)
     idfDict = dict.fromkeys(documents[0].keys(),0)
     for document in documents:
       for word, val in document.items():
          if val > 0 :
            idfDict[word] += 1
     for word, val in idfDict.items():
      idfDict[word] = math.log(N / float(val))
     return idfDict
idfs = computeIDF([numOfWordsA,numOfWordsB])
idfs
```

Out[18]:
```
{'Sun': 0.6931471805599453,
 'the': 0.0,
 'is': 0.0,
 'fourth': 0.6931471805599453,
 'largest': 0.6931471805599453,
 'Planet': 0.6931471805599453,
 'from': 0.6931471805599453,
 'Mars': 0.6931471805599453,
 'planet': 0.6931471805599453,
 'Jupiter': 0.6931471805599453}
```

In [182…
```
def computeTFIDF(tfBagOfWords, idfs):
    tfidf = {}
    for word, val in tfBagOfWords.items():
      tfidf[word] = val * idfs[word]
    return tfidf
tfidfA = computeTFIDF(tfA,idfs)
tfidfB = computeTFIDF(tfB,idfs)
df = pd.DataFrame([tfidfA,tfidfB])
df
```

Out[182…

	is	Planet	Jupiter	the	planet	Mars	fourth	Sun	from	largest
0	0.0	0.138629	0.138629	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.138629
1	0.0	0.000000	0.000000	0.0	0.086643	0.086643	0.086643	0.086643	0.086643	0.000000

In [ ]:
```
'''
Name: Rohan Dhadke
```

```
TE-A2
'''
```