In [1]:
```
import nltk
from nltk import word_tokenize
text="We need to Tokenize this text and perform the given Activities"
```

In [2]:
```
#lower case
text=text.lower()
print(text)
```

```
we need to tokenize this text and perform the given activities
```

In [3]:
```
#Tokenize
print(nltk.word_tokenize(text))
```

```
['we', 'need', 'to', 'tokenize', 'this', 'text', 'and', 'perform', 'the', 'given', 'activities']
```

In [4]:
```
#StopWords Removal

from nltk.corpus import stopwords
stop_word= set(stopwords.words('english'))
words=word_tokenize(text)
filtered_words=[word for word in words if word.lower() not in stop_word]

filtered_text=" ".join(filtered_words)
print(filtered_text)
```

```
need tokenize text perform given activities
```

In [5]:
```
# Stemming
from nltk.stem import PorterStemmer
porter=PorterStemmer()
print(porter.stem(text))
```

```
we need to tokenize this text and perform the given act
```

In [6]:
```
#Lemmatizing
import nltk
nltk.download("wordnet")
from nltk.stem import WordNetLemmatizer
lemmatizer=WordNetLemmatizer()

text=nltk.word_tokenize(text)
filtered_words=word_tokenize(filtered_text)
lemmatized_words=[lemmatizer.lemmatize(word, pos='v') for word in filtered_words]
filtered_text=" ".join(filtered_words)
print(filtered_text)
```

```
[nltk_data] Downloading package wordnet to /home/ubuntu/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
need tokenize text perform given activities
```

In [9]:
```
#tf/idf
import re
import string
# assign documents
d0 = 'This is document 1'
d1 = 'Document 2'
d2 = 'and Document 3'

# merge documents into a single corpus
string = [d0, d1, d2]
# import required module
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [10]:
```
# create object
tfidf = TfidfVectorizer()

# get tf-df values
result = tfidf.fit_transform(string)
```

In [11]:
```
# get indexing
print('\nWord indexes:')
print(tfidf.vocabulary_)

# display tf-idf values
print('\ntf-idf value:')
print(result)

# in matrix form
print('\ntf-idf values in matrix form:')
print(result.toarray())
```

```
Word indexes:
{'this': 3, 'is': 2, 'document': 1, 'and': 0}

tf-idf value:
  (0, 1)        0.3853716274664007
  (0, 2)        0.652490884512534
  (0, 3)        0.652490884512534
  (1, 1)        1.0
  (2, 0)        0.8610369959439764
  (2, 1)        0.5085423203783267

tf-idf values in matrix form:
[[0.         0.38537163 0.65249088 0.65249088]
 [0.         1.         0.         0.        ]
 [0.861037   0.50854232 0.         0.        ]]
```

In [ ]:
```
```