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

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

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

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

```
---------------------------------------------------------------------------
LookupError                               Traceback (most recent call last)
/tmp/ipykernel_7280/758391351.py in <module>
----> 1 print(nltk.word_tokenize(text))

~/.local/lib/python3.10/site-packages/nltk/tokenize/__init__.py in word_tokenize(text, language, preserve_line)
    127     :type preserve_line: bool
    128     """
--> 129     sentences = [text] if preserve_line else sent_tokenize(text, language)
    130     return [
    131         token for sent in sentences for token in _treebank_word_tokenizer.tokenize(sent)

~/.local/lib/python3.10/site-packages/nltk/tokenize/__init__.py in sent_tokenize(text, language)
    104     :param language: the model name in the Punkt corpus
    105     """
--> 106     tokenizer = load(f"tokenizers/punkt/{language}.pickle")
    107     return tokenizer.tokenize(text)
    108

~/.local/lib/python3.10/site-packages/nltk/data.py in load(resource_url, format, cache, verbose, logic_parser, fstruct_reader, encoding)
    748
    749     # Load the resource.
--> 750     opened_resource = _open(resource_url)
    751
    752     if format == "raw":

~/.local/lib/python3.10/site-packages/nltk/data.py in _open(resource_url)
    874
    875     if protocol is None or protocol.lower() == "nltk":
--> 876         return find(path_, path + [""]).open()
    877     elif protocol.lower() == "file":
    878         # urllib might not use mode='rb', so handle this one ourselves:

~/.local/lib/python3.10/site-packages/nltk/data.py in find(resource_name, paths)
    581     sep = "*" * 70
    582     resource_not_found = f"\n{sep}\n{msg}\n{sep}\n"
--> 583     raise LookupError(resource_not_found)
    584
    585

LookupError:
**********************************************************************
  Resource punkt not found.
  Please use the NLTK Downloader to obtain the resource:

  >>> import nltk
  >>> nltk.download('punkt')

  For more information see: https://www.nltk.org/data.html

  Attempted to load tokenizers/punkt/PY3/english.pickle

  Searched in:
    - '/home/ubuntu/nltk_data'
    - '/usr/nltk_data'
    - '/usr/share/nltk_data'
    - '/usr/lib/nltk_data'
    - '/usr/share/nltk_data'
    - '/usr/local/share/nltk_data'
    - '/usr/lib/nltk_data'
    - '/usr/local/lib/nltk_data'
    - ''
**********************************************************************
```

In [4]:
```
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /home/ubuntu/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
```

Out[4]:
```
True
```

In [5]:
```
print(nltk.word_tokenize(text))
```

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

In [6]:
```
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)
```

```
---------------------------------------------------------------------------
LookupError                               Traceback (most recent call last)
~/.local/lib/python3.10/site-packages/nltk/corpus/util.py in __load(self)
     83                 try:
---> 84                     root = nltk.data.find(f"{self.subdir}/{zip_name}")
     85                 except LookupError:

~/.local/lib/python3.10/site-packages/nltk/data.py in find(resource_name, paths)
    582     resource_not_found = f"\n{sep}\n{msg}\n{sep}\n"
--> 583     raise LookupError(resource_not_found)
    584

LookupError:
**********************************************************************
  Resource stopwords not found.
  Please use the NLTK Downloader to obtain the resource:

  >>> import nltk
  >>> nltk.download('stopwords')

  For more information see: https://www.nltk.org/data.html

  Attempted to load corpora/stopwords.zip/stopwords/

  Searched in:
    - '/home/ubuntu/nltk_data'
    - '/usr/nltk_data'
    - '/usr/share/nltk_data'
    - '/usr/lib/nltk_data'
    - '/usr/share/nltk_data'
    - '/usr/local/share/nltk_data'
    - '/usr/lib/nltk_data'
    - '/usr/local/lib/nltk_data'
**********************************************************************


During handling of the above exception, another exception occurred:

LookupError                               Traceback (most recent call last)
/tmp/ipykernel_7280/747462371.py in <module>
      1 from nltk.corpus import stopwords
----> 2 stop_word= set(stopwords.words('english'))
      3 words=word_tokenize(text)
      4 filtered_words=[word for word in words if word.lower() not in stop_word]
      5

~/.local/lib/python3.10/site-packages/nltk/corpus/util.py in __getattr__(self, attr)
    119             raise AttributeError("LazyCorpusLoader object has no attribute '__bases__'")
    120
--> 121         self.__load()
    122         # This looks circular, but its not, since __load() changes our
    123         # __class__ to something new:

~/.local/lib/python3.10/site-packages/nltk/corpus/util.py in __load(self)
     84                     root = nltk.data.find(f"{self.subdir}/{zip_name}")
     85                 except LookupError:
---> 86                     raise e
     87
     88         # Load the corpus.

~/.local/lib/python3.10/site-packages/nltk/corpus/util.py in __load(self)
     79         else:
     80             try:
---> 81                 root = nltk.data.find(f"{self.subdir}/{self.__name}")
     82             except LookupError as e:
     83                 try:

~/.local/lib/python3.10/site-packages/nltk/data.py in find(resource_name, paths)
    581     sep = "*" * 70
    582     resource_not_found = f"\n{sep}\n{msg}\n{sep}\n"
--> 583     raise LookupError(resource_not_found)
    584
    585

LookupError:
**********************************************************************
  Resource stopwords not found.
  Please use the NLTK Downloader to obtain the resource:

  >>> import nltk
  >>> nltk.download('stopwords')

  For more information see: https://www.nltk.org/data.html

  Attempted to load corpora/stopwords

  Searched in:
    - '/home/ubuntu/nltk_data'
    - '/usr/nltk_data'
    - '/usr/share/nltk_data'
    - '/usr/lib/nltk_data'
    - '/usr/share/nltk_data'
    - '/usr/local/share/nltk_data'
    - '/usr/lib/nltk_data'
    - '/usr/local/lib/nltk_data'
**********************************************************************
```

In [7]:
```
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /home/ubuntu/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

Out[7]:
```
True
```

In [8]:
```
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 [9]:
```
from nltk.stem import PorterStemmer
porter=PorterStemmer()
print(porter.stem(text))
```

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

In [10]:
```
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...
need tokenize text perform given activities
```

In [11]:
```
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 [12]:
```
tfidf = TfidfVectorizer()
```

In [13]:
```
result = tfidf.fit_transform(string)
```

In [14]:
```
# 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 [ ]: