In [1]:

```
import nltk
```

In [2]:

```
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\user\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

Out[2]:

```
True
```

In [3]:

```
from nltk.tokenize import sent_tokenize,word_tokenize
```

In [47]:

```
text="""Data science and big data analytics is use to visualize the data.consist
```

In [48]:

```
tokenized_sent=sent_tokenize(text)
print(tokenized_sent)
```

```
['Data science and big data analytics is use to visualize the data.consist of p
rediction and clustering.', 'only one piece.been']
```

In [49]:

```
tokens=word_tokenize(text)
print(tokens)
```

```
['Data', 'science', 'and', 'big', 'data', 'analytics', 'is', 'use', 'to', 'visu
alize', 'the', 'data.consist', 'of', 'prediction', 'and', 'clustering', '.', 'o
nly', 'one', 'piece.been']
```

In [50]:

```
from nltk.corpus import stopwords
```

In [51]:

```
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\user\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[51]:

```
True
```

In [52]:
```
stop_words=set(stopwords.words("english"))
print(stop_words)
```

```
{'isn', 'further', 're', 't', 'what', 'up', 'has', 'all', "don't", 'can', 'do
n', "couldn't", 'an', 'most', 'hasn', "won't", 'i', "haven't", 'after', 'too',
'between', 'yours', 'how', 'mightn', 'yourselves', "weren't", 'themselves', 'th
at', "aren't", 'm', 'had', 'her', 'she', 'do', 'as', "isn't", "shouldn't", 'oth
er', 'below', 'have', 'over', 'under', 'their', 'they', 'against', 'not', 'hers
elf', 'any', 'now', 'whom', 'does', "mightn't", 'doing', 'why', 'for', "has
n't", 'aren', "needn't", 'did', 'by', 'but', "you'd", 'haven', 'wasn', "you'l
l", 'about', 'both', 'him', "she's", 'and', 'shouldn', 'of', 'his', 've', 'int
o', "should've", 'who', 'be', 'more', 'those', 'he', 'was', 'during', 'so', 'th
e', 'out', 'o', 'we', 's', 'should', 'here', 'will', 'am', 'll', 'were', 'our
s', 'these', 'to', 'my', 'above', 'if', 'just', "hadn't", 'from', 'himself', 'o
r', 'few', 'own', "doesn't", 'off', 'down', 'each', 'ain', 'because', 'doesn',
"it's", 'wouldn', 'ourselves', 'is', 'some', 'this', 'couldn', 'didn', 'me', 'i
n', 'mustn', 'your', 'again', 'once', 'nor', "didn't", 'having', 'theirs', 'wer
en', "mustn't", 'there', 'at', 'are', "wasn't", 'then', 'with', "wouldn't", 'be
ing', 'than', 'no', 'until', 'very', 'needn', 'which', 'same', 'hers', 'on', "y
ou've", 'when', 'where', 'before', 'd', "that'll", 'ma', 'y', 'such', 'our', 'o
nly', "shan't", 'yourself', 'itself', 'its', "you're", 'hadn', 'them', 'a', 'my
self', 'you', 'while', 'it', 'through', 'won', 'been', 'shan'}
```

In [53]:
```
filtered_sent=[]
for w in tokenized_sent:
    if w not in stop_words:
        filtered_sent.append(w)
print("Tokenized Sentence:",tokenized_sent)
print("Filterd Sentence:",filtered_sent)
```

```
Tokenized Sentence: ['Data science and big data analytics is use to visualize t
he data.consist of prediction and clustering.', 'only one piece.been']
Filterd Sentence: ['Data science and big data analytics is use to visualize the
data.consist of prediction and clustering.', 'only one piece.been']
```

In [54]:
```
from nltk.stem import PorterStemmer
from nltk.tokenize import sent_tokenize, word_tokenize

ps = PorterStemmer()

stemmed_words=[]
for w in filtered_sent:
    stemmed_words.append(ps.stem(w))

print("Filtered Sentence:",filtered_sent)
print("Stemmed Sentence:",stemmed_words)
```

```
Filtered Sentence: ['Data science and big data analytics is use to visualize th
e data.consist of prediction and clustering.', 'only one piece.been']
Stemmed Sentence: ['data science and big data analytics is use to visualize the
data.consist of prediction and clustering.', 'only one piece.been']
```

In [55]:

```python
nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\user\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

Out[55]:

```
True
```

In [56]:

```python
from nltk.stem.wordnet import WordNetLemmatizer
lem = WordNetLemmatizer()

from nltk.stem.porter import PorterStemmer
stem = PorterStemmer()

word = "flying"
print("Lemmatized Word:",lem.lemmatize(word,"v"))
print("Stemmed Word:",stem.stem(word))
```

```
Lemmatized Word: fly
Stemmed Word: fli
```

In [14]:

```python
nltk.download('averaged_perceptron_tagger')
```

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\user\AppData\Roaming\nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
```

Out[14]:

```
True
```

In [57]:

```python
nltk.pos_tag(tokens)
```

Out[57]:

```
[('Data', 'NNP'),
 ('science', 'NN'),
 ('and', 'CC'),
 ('big', 'JJ'),
 ('data', 'NNS'),
 ('analytics', 'NNS'),
 ('is', 'VBZ'),
 ('use', 'JJ'),
 ('to', 'TO'),
 ('visualize', 'VB'),
 ('the', 'DT'),
 ('data.consist', 'NN'),
 ('of', 'IN'),
 ('prediction', 'NN'),
 ('and', 'CC'),
 ('clustering', 'NN'),
 ('.', '.'),
 ('only', 'RB'),
 ('one', 'CD'),
 ('piece.been', 'NN')]
```

In [58]:

```
import pandas as pd
import numpy as np
```

In [59]:

```
# import required module
from sklearn.feature_extraction.text import TfidfVectorizer

# assign documents
d0 = 'hrutika jare'
d1 = 'rutuja jarange'


# merge documents into a single corpus
string = [d0, d1]

# create object
tfidf = TfidfVectorizer()

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

# get indexing
print('\nWord indexes:')
print(tfidf.vocabulary_)

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

```

Word indexes:
{'hrutika': 0, 'jare': 2, 'rutuja': 3, 'jarange': 1}

tf-idf values:
  (0, 2)	0.7071067811865476
  (0, 0)	0.7071067811865476
  (1, 1)	0.7071067811865476
  (1, 3)	0.7071067811865476
```

In [ ]:

```

```