```
import nltk
nltk.download ()
```

```
NLTK Downloader
---------------------------------------------------------------------------
    d) Download   l) List    u) Update   c) Config   h) Help   q) Quit
---------------------------------------------------------------------------
Downloader> q
True
```

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

```
nltk.download('punkt')
data="Recent times have witnessed an explosion in the amount of biological data generated. There are millions of research ar

#STEP 1 :TOKENIZATION : Breaking complex data into simple units
#Sentence Tokenizer
sentences=sent_tokenize(data)
print(sentences)
```

```
#Word Tokenizer
words=word_tokenize(data)
print(words)
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
['Recent times have witnessed an explosion in the amount of biological data generated.', 'There are millions of resear
['Recent', 'times', 'have', 'witnessed', 'an', 'explosion', 'in', 'the', 'amount', 'of', 'biological', 'data', 'genera
```

```
nltk.download('stopwords')
from nltk.corpus import stopwords
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

```
#Data Cleaning :Remove Stopwords and Punctuations

import re
from nltk.corpus import stopwords
data="Coronavirus disease is an infectious disease caused by a newly discovered coronavirus. Most people infected with the C
data = re.sub('[^a-zA-Z]', ' ',data)
data = data.lower()
data = data.split()
data = [word for word in data if not word in set(stopwords.words('english'))]
data = ' '.join(data)
print(data)
```

```
coronavirus disease infectious disease caused newly discovered coronavirus people infected covid virus experience mild
```

```
#STEP 2: STEMMING
#Create object of PorterStemmer
from nltk.stem import PorterStemmer
stemmer=PorterStemmer()
for i in range(len(sentences)):
    words=word_tokenize(sentences[i])
    #List comprehension
    words=[stemmer.stem(word) for word in words if word not in set(stopwords.words('english'))]
    sentences[i]=' '.join(words)
print(sentences)
```

```
['recent time wit explos amount biolog data gener .', 'there million research articl pivot inform human health diseas
```

```
from nltk.stem import WordNetLemmatizer
nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

```
True
```

```
#STEP 3: LEMMATIZATION
#Stemming with lemmatization to get proper meaning words after stemming
#Create obj of Lemmatizer

lemmmatizer=WordNetLemmatizer()

for i in range(len(sentences)):
    words=word_tokenize(sentences[i])
    #List comprehension
    words = [lemmmatizer.lemmatize(word.lower()) for word in words if word not in set(stopwords.words('english'))]
    sentences[i]=' '.join(words)
print(sentences)
```

```
['recent time wit explos amount biolog data gener .', 'million research articl pivot inform human health diseas , span
```

```
#Step 3 : Bag of Words : Document Matrix , Count Vectorizer
from sklearn.feature_extraction.text import CountVectorizer

cv=CountVectorizer()
x=cv.fit_transform(sentences).toarray()
print(x)
```

```
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
```

```
#TF-IDF
from sklearn.feature_extraction.text import TfidfVectorizer
cv=TfidfVectorizer()
x=cv.fit_transform(sentences).toarray()
print(x)
```

```
[[0.         0.         0.         ... 0.         0.         0.        ]
 [0.         0.         0.         ... 0.         0.         0.        ]
 [0.         0.         0.         ... 0.         0.         0.        ]
 ...
 [0.         0.         0.         ... 0.         0.28433499 0.        ]
 [0.         0.         0.         ... 0.         0.         0.        ]
 [0.         0.         0.         ... 0.         0.         0.        ]]
```

# word2vec

```
import pandas as pd
df=pd.read_csv("/content/spam.csv",encoding='ISO-8859-1')
df.head()
```

| | v1 | v2 | Unnamed: 2 | Unnamed: 3 | Unnamed: 4 |
|---|---|---|---|---|---|
| **0** | ham | Go until jurong point, crazy.. Available only ... | NaN | NaN | NaN |
| **1** | ham | Ok lar... Joking wif u oni... | NaN | NaN | NaN |
| **2** | spam | Free entry in 2 a wkly comp to win FA Cup fina... | NaN | NaN | NaN |
| **3** | ham | U dun say so early hor... U c already then say... | NaN | NaN | NaN |
| **4** | ham | Nah I don't think he goes to usf, he lives aro... | NaN | NaN | NaN |

```
a=df['v2']
a.head(30)
```

```
0     Go until jurong point, crazy.. Available only ...
1                         Ok lar... Joking wif u oni...
```

```
2     Free entry in 2 a wkly comp to win FA Cup fina...
3     U dun say so early hor... U c already then say...
4     Nah I don't think he goes to usf, he lives aro...
5     FreeMsg Hey there darling it's been 3 week's n...
6     Even my brother is not like to speak with me. ...
7     As per your request 'Melle Melle (Oru Minnamin...
8     WINNER!! As a valued network customer you have...
9     Had your mobile 11 months or more? U R entitle...
10    I'm gonna be home soon and i don't want to tal...
11    SIX chances to win CASH! From 100 to 20,000 po...
12    URGENT! You have won a 1 week FREE membership ...
13    I've been searching for the right words to tha...
14                  I HAVE A DATE ON SUNDAY WITH WILL!!
15    XXXMobileMovieClub: To use your credit, click ...
16                           Oh k...i'm watching here:)
17    Eh u remember how 2 spell his name... Yes i di...
18    Fine if thatåÕs the way u feel. ThatåÕs the wa...
19    England v Macedonia - dont miss the goals/team...
20           Is that seriously how you spell his name?
21    I□Û÷m going to try for 2 months ha ha only joking
22    So Ì_ pay first lar... Then when is da stock c...
23    Aft i finish my lunch then i go str down lor. ...
24    Fffffffff. Alright no way I can meet up with ...
25    Just forced myself to eat a slice. I'm really ...
26                       Lol your always so convincing.
27    Did you catch the bus ? Are you frying an egg ...
28    I'm back &amp;amp; we're packing the car now, I'll...
29    Ahhh. Work. I vaguely remember that! What does...
Name: v2, dtype: object
```

```
newValue=[word_tokenize(i) for i in a]
print(newValue)
```

```
urong', 'point', ',', 'crazy..', 'Available', 'only', 'in', 'bugis', 'n', 'great', 'world', 'la', 'e', 'buffet', '...',
```

```
model=Word2Vec(newValue,min_count=1,size=32) # min_count=1 represents to consider the word even if it occurs once
print(model)
```

```
Word2Vec(vocab=11899, size=32, alpha=0.025)
```

```
model.most_similar('mobile')
```

```
[('Call', 0.9997293949127197),
 ('per', 0.999592125415802),
 ('cash', 0.9994790554046631),
 ('Box', 0.9994083642959595),
 ('18', 0.9993181228637695),
 ('Statement', 0.9992611408233643),
 ('or', 0.9992474317550659),
 ('Nokia', 0.9992243647575378),
 ('mins', 0.9991412162780762),
 ('PO', 0.9991174936294556)]
```

```
bruce='brue banner is a scientist. He is an Strogest Avenger'
s_b=sent_tokenize(bruce)
print(s_b)
w_b=[word_tokenize(i) for i in s_b]
print(w_b)
```

```
['brue banner is a scientist.', 'He is an Strogest Avenger']
[['brue', 'banner', 'is', 'a', 'scientist', '.'], ['He', 'is', 'an', 'Strogest', 'Avenger']]
```

```
model=Word2Vec(w_b,min_count=1)
print(model)
```

```
Word2Vec(vocab=10, size=100, alpha=0.025)
```

```
model.most_similar('brue')
```

```
[('Strogest', 0.16545388102531433),
 ('a', 0.11724375933408737),
 ('He', 0.08706473559141159),
 ('banner', 0.04554155468940735),
 ('Avenger', 0.031042540445923805),
 ('an', 0.015547312796115875),
 ('scientist', -0.03471317142248154),
 ('.', -0.03711579367518425),
 ('is', -0.1317368894815445)]
```