

```
!pip install nltk
import nltk
nltk.download('plunkt')
nltk.download('wordnet')
nltk.download('averaged_perceptron_tagger')
nltk.download('stopwords')
from nltk import sent_tokenize
from nltk import word_tokenize
from nltk.corpus import stopwords
```

Defaulting to user installation because normal site-packages is not writeable

Collecting nltk

Downloading nltk-3.9.1-py3-none-any.whl.metadata (2.9 kB)

Requirement already satisfied: click in c:\users\mahesh dhanve\appdata\roaming\python\python311\site-packages (from nltk) (8.1.7)

Requirement already satisfied: joblib in c:\users\mahesh dhanve\appdata\roaming\python\python311\site-packages (from nltk) (1.4.2)

Collecting regex>=2021.8.3 (from nltk)

Downloading regex-2024.11.6-cp311-cp311-win\_amd64.whl.metadata (41 kB)

Collecting tqdm (from nltk)

Downloading tqdm-4.67.1-py3-none-any.whl.metadata (57 kB)

Requirement already satisfied: colorama in c:\users\mahesh dhanve\appdata\roaming\python\python311\site-packages (from click->nltk) (0.4.6)

Downloading nltk-3.9.1-py3-none-any.whl (1.5 MB)

```
----- 0.0/1.5 MB ? eta -:--:--
----- 0.3/1.5 MB ? eta -:--:--
----- 0.3/1.5 MB ? eta -:--:--
----- 0.5/1.5 MB 837.5 kB/s eta
0:00:02
----- 0.8/1.5 MB 860.9 kB/s eta
0:00:01
----- 1.0/1.5 MB 915.5 kB/s eta
0:00:01
----- 1.3/1.5 MB 986.4 kB/s eta
0:00:01
----- 1.5/1.5 MB 970.6 kB/s eta
0:00:00
```

Downloading regex-2024.11.6-cp311-cp311-win\_amd64.whl (274 kB)

Downloading tqdm-4.67.1-py3-none-any.whl (78 kB)

Installing collected packages: tqdm, regex, nltk

Successfully installed nltk-3.9.1 regex-2024.11.6 tqdm-4.67.1

[notice] A new release of pip is available: 24.2 -> 25.0.1

[notice] To update, run: python.exe -m pip install --upgrade pip

[nltk\_data] Error loading plunkt: Package 'plunkt' not found in index

[nltk\_data] Downloading package wordnet to C:\Users\Mahesh

```
[nltk_data]      Dhanve\AppData\Roaming\nltk_data...
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]      C:\Users\Mahesh Dhanve\AppData\Roaming\nltk_data...
[nltk_data] Unzipping taggers\averaged_perceptron_tagger.zip.
[nltk_data] Downloading package stopwords to C:\Users\Mahesh
[nltk_data]      Dhanve\AppData\Roaming\nltk_data...
[nltk_data] Unzipping corpora\stopwords.zip.
```

```
text='Real madrid is set to win the UCL for the season . Benzema might
win Balon dor . Salah might be the runner up'
```

```
pip install --upgrade nltk
```

```
Defaulting to user installation because normal site-packages is not
writeable
```

```
Requirement already satisfied: nltk in c:\users\mahesh dhanve\appdata\
roaming\python\python311\site-packages (3.9.1)
```

```
Requirement already satisfied: click in c:\users\mahesh dhanve\
appdata\roaming\python\python311\site-packages (from nltk) (8.1.7)
```

```
Requirement already satisfied: joblib in c:\users\mahesh dhanve\
appdata\roaming\python\python311\site-packages (from nltk) (1.4.2)
```

```
Requirement already satisfied: regex<=2021.8.3 in c:\users\mahesh
dhanve\appdata\roaming\python\python311\site-packages (from nltk)
(2024.11.6)
```

```
Requirement already satisfied: tqdm in c:\users\mahesh dhanve\appdata\
roaming\python\python311\site-packages (from nltk) (4.67.1)
```

```
Requirement already satisfied: colorama in c:\users\mahesh dhanve\
appdata\roaming\python\python311\site-packages (from click->nltk)
(0.4.6)
```

```
Note: you may need to restart the kernel to use updated packages.
```

```
[notice] A new release of pip is available: 24.2 -> 25.0.1
```

```
[notice] To update, run: python.exe -m pip install --upgrade pip
```

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

```
[nltk_data] Downloading package punkt to C:\Users\Mahesh
```

```
[nltk_data]      Dhanve\AppData\Roaming\nltk_data...
```

```
[nltk_data] Package punkt is already up-to-date!
```

```
True
```

```
nltk.download('punkt_tab')
```

```
text='Real madrid is set to win the UCL for the season . Benzema might
win Balon dor . Salah might be the runner up'
```

```
tokens_sents = nltk.sent_tokenize(text)
```

```
print(tokens_sents)
```

```

[nltk_data] Downloading package punkt_tab to C:\Users\Mahesh
[nltk_data]      Dhanve\AppData\Roaming\nltk_data...

['Real madrid is set to win the UCL for the season .', 'Benzema might
win Balon dor .', 'Salah might be the runner up']

[nltk_data]   Unzipping tokenizers\punkt_tab.zip.

tokens_words = nltk.word_tokenize(text)
print(tokens_words)

['Real', 'madrid', 'is', 'set', 'to', 'win', 'the', 'UCL', 'for',
'the', 'season', '.', 'Benzema', 'might', 'win', 'Balon', 'dor', '.',
'Salah', 'might', 'be', 'the', 'runner', 'up']

from nltk.stem import PorterStemmer
from nltk.stem.snowball import SnowballStemmer
from nltk.stem import LancasterStemmer

stem=[]
for i in tokens_words:
    ps = PorterStemmer()
    stem_word= ps.stem(i)
    stem.append(stem_word)
print(stem)

['real', 'madrid', 'is', 'set', 'to', 'win', 'the', 'ucl', 'for',
'the', 'season', '.', 'benzema', 'might', 'win', 'balon', 'dor', '.',
'salah', 'might', 'be', 'the', 'runner', 'up']

# Lemmatization

import nltk
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()

lemmatized_output = ' '.join([lemmatizer.lemmatize(w) for w in stem])
print(lemmatized_output)

real madrid is set to win the ucl for the season . benzema might win
balon dor . salah might be the runner up

leme=[]
for i in stem:
    lemetized_word=lemmatizer.lemmatize(i)
    leme.append(lemetized_word)
print(leme)

['real', 'madrid', 'is', 'set', 'to', 'win', 'the', 'ucl', 'for',
'the', 'season', '.', 'benzema', 'might', 'win', 'balon', 'dor', '.',
'salah', 'might', 'be', 'the', 'runner', 'up']

```

### #PART OF SPEECH TAGGING

```
nltk.download('averaged_perceptron_tagger_eng')
print("Parts of Speech: ",nltk.pos_tag(leme))
```

```
[nltk_data] Downloading package averaged_perceptron_tagger_eng to
[nltk_data] C:\Users\Mahesh Dhanve\AppData\Roaming\nltk_data...
[nltk_data] Unzipping taggers\averaged_perceptron_tagger_eng.zip.
```

```
Parts of Speech: [('real', 'JJ'), ('madrid', 'NN'), ('is', 'VBZ'),
('set', 'VBN'), ('to', 'TO'), ('win', 'VB'), ('the', 'DT'), ('ucl',
'NN'), ('for', 'IN'), ('the', 'DT'), ('season', 'NN'), ('.', '.'),
('benzema', 'NN'), ('might', 'MD'), ('win', 'VB'), ('balon', 'NN'),
('dor', 'NN'), ('.', '.'), ('salah', 'NN'), ('might', 'MD'), ('be',
'VB'), ('the', 'DT'), ('runner', 'NN'), ('up', 'RP')]
```

### # STOP WORD

```
from nltk.corpus import stopwords
sw_nltk = stopwords.words('english')
print(sw_nltk)
```

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

```
words = [word for word in text.split() if word.lower() not in sw_nltk]
new_text = " ".join(words)
print(new_text)
```

Real madrid set win UCL season . Benzema might win Balon dor . Salah might runner

#### *# POS TAGGING*

```
sent = " "  
tokens=nltk.word_tokenize(sent)  
print(tokens)  
['Albert', 'Einstein', 'was', 'born', 'in', 'Ulm', ',', 'Germany',  
'in', '1879', '.']  
nltk.pos_tag(tokens)
```

```
[]
```

```
[]
```

#### *# Stemming*

```
import nltk  
from nltk.corpus import stopwords  
from nltk.tokenize import word_tokenize  
from nltk.stem import PorterStemmer
```

#### *# Download necessary NLTK data files*

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

#### *# Sample text*

```
text = "NLTK is a powerful library for processing human language  
data."
```

#### *# Tokenize into words*

```
words = word_tokenize(text)
```

#### *# Remove stopwords*

```
stop_words = set(stopwords.words('english'))  
filtered_sent = [w for w in words if w.lower() not in stop_words]
```

#### *# Initialize stemmer*

```
ps = PorterStemmer()
```

#### *# Stem the words*

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

#### *# Output*

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

```
Filtered Sentence: ['NLTK', 'powerful', 'library', 'processing',  
'human', 'language', 'data', '.']
```

```
Stemmed Sentence: ['nltk', 'power', 'librari', 'process', 'human',  
'languag', 'data', '.']
```

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