```
#sample text

sample_text = "The quick brown fox jumps over the lazy dog. Dogs are
not lazy by nature"

# import nltk with important download
import nltk
import pandas as pd
nltk.download('punkt')
nltk.download('punkt_tab')
nltk.download('averaged_perceptron_tagger')
nltk.download('averaged_perceptron_tagger_eng')
nltk.download('stopwords')
nltk.download('wordnet')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Lenovo\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package punkt_tab to
[nltk_data]     C:\Users\Lenovo\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt_tab is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\Lenovo\AppData\Roaming\nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
[nltk_data] Downloading package averaged_perceptron_tagger_eng to
[nltk_data]     C:\Users\Lenovo\AppData\Roaming\nltk_data...
[nltk_data]   Package averaged_perceptron_tagger_eng is already up-to-
[nltk_data]       date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Lenovo\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\Lenovo\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

```
True
```

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

# tokenization


tokens = word_tokenize(sample_text)
print("Tokens: " , tokens)
#pos tagging
```

```
pos_tags = pos_tag(tokens)
print("\nPOS Tags: ", pos_tags)


#Remove stopwords

stop_words = set(stopwords.words('english'))
filtered_tokens = [word for word in tokens if word.lower() not in
stop_words]
print("\n Filtered tokens (StopWords Removed): ", filtered_tokens)

#stemming

stemmer = PorterStemmer()
stemmed_tokens  = [stemmer.stem(word) for word in filtered_tokens]
print("\nStemmed tokens:",stemmed_tokens)

#lemmatization

lemmatizer = WordNetLemmatizer()
lemmatized_tokens = [lemmatizer.lemmatize(word) for word in
filtered_tokens]
print("\nLemmatized tokens: ", lemmatized_tokens)
```

```
Tokens:  ['The', 'quick', 'brown', 'fox', 'jumps', 'over', 'the',
'lazy', 'dog', '.', 'Dogs', 'are', 'not', 'lazy', 'by', 'nature']

POS Tags:  [('The', 'DT'), ('quick', 'JJ'), ('brown', 'NN'), ('fox',
'NN'), ('jumps', 'VBZ'), ('over', 'IN'), ('the', 'DT'), ('lazy',
'JJ'), ('dog', 'NN'), ('.', '.'), ('Dogs', 'NNS'), ('are', 'VBP'),
('not', 'RB'), ('lazy', 'JJ'), ('by', 'IN'), ('nature', 'NN')]

 Filtered tokens (StopWords Removed):  ['quick', 'brown', 'fox',
'jumps', 'lazy', 'dog', '.', 'Dogs', 'lazy', 'nature']

Stemmed tokens: ['quick', 'brown', 'fox', 'jump', 'lazi', 'dog', '.',
'dog', 'lazi', 'natur']

Lemmatized tokens:  ['quick', 'brown', 'fox', 'jump', 'lazy', 'dog',
'.', 'Dogs', 'lazy', 'nature']
```

```
# term frequency and inverse document frequency(IDF)

from sklearn.feature_extraction.text import TfidfVectorizer
# sample corpus(multiple document)

corpus = [
    "The quick brown fox.",
    "The lazy dog sleeps.",
    "Dogs and foxes are animals."
]
```

```
# TF-IDF Vecotorizer
vectorizer = TfidfVectorizer()

#fit and transform the corpus

tfidf_matrix = vectorizer.fit_transform(corpus)

# convert the tf-idf matrix to a readable format

feature_names = vectorizer.get_feature_names_out()
dense  = tfidf_matrix.todense()
df_tfidf = pd.DataFrame(dense,columns = feature_names)

print("\n TF-IDF Matrix:")
print(df_tfidf)
```

```
 TF-IDF Matrix:
        and   animals       are     brown       dog      dogs
fox  \
0  0.000000  0.000000  0.000000  0.528635  0.000000  0.000000
0.528635
1  0.000000  0.000000  0.000000  0.000000  0.528635  0.000000
0.000000
2  0.447214  0.447214  0.447214  0.000000  0.000000  0.447214
0.000000

      foxes      lazy     quick    sleeps      the
0  0.000000  0.000000  0.528635  0.000000  0.40204
1  0.000000  0.528635  0.000000  0.528635  0.40204
2  0.447214  0.000000  0.000000  0.000000  0.00000
```