```
Lab Assignment 7
Aim : 1. Extract Sample document and apply following document
preprocessing methods:
Tokenization, POS Tagging, stop words removal, Stemming and
Lemmatization.
2. Create representation of document by calculating Term Frequency and
Inverse Document
Frequency

#Tokenization

from nltk import word_tokenize, sent_tokenize

corpus = "Sachin was the GOAT of the previous generation. Virat is the
GOAT of this generation. Shubman will be the GOAT of the next
generation"

print(word_tokenize(corpus))
print(sent_tokenize(corpus))
```

```
['Sachin', 'was', 'the', 'GOAT', 'of', 'the', 'previous',
'generation', '.', 'Virat', 'is', 'the', 'GOAT', 'of', 'this',
'generation', '.', 'Shubman', 'will', 'be', 'the', 'GOAT', 'of',
'the', 'next', 'generation']
['Sachin was the GOAT of the previous generation.', 'Virat is the GOAT
of this generation.', 'Shubman will be the GOAT of the next
generation']
```

```
#POS Tagging

from nltk import pos_tag

tokens = word_tokenize(corpus)
print(pos_tag(tokens))
```

```
[('Sachin', 'NNP'), ('was', 'VBD'), ('the', 'DT'), ('GOAT', 'NNP'),
('of', 'IN'), ('the', 'DT'), ('previous', 'JJ'), ('generation', 'NN'),
('.', '.'), ('Virat', 'NNP'), ('is', 'VBZ'), ('the', 'DT'), ('GOAT',
'NNP'), ('of', 'IN'), ('this', 'DT'), ('generation', 'NN'), ('.',
'.'), ('Shubman', 'NNP'), ('will', 'MD'), ('be', 'VB'), ('the', 'DT'),
('GOAT', 'NNP'), ('of', 'IN'), ('the', 'DT'), ('next', 'JJ'),
('generation', 'NN')]
```

```
#Stop word removal

from nltk.corpus import stopwords
stop_words = set(stopwords.words("english"))

tokens = word_tokenize(corpus)
cleaned_tokens = []
for token in tokens:
  if (token not in stop_words):
```

```
    cleaned_tokens.append(token)
print(cleaned_tokens)
```

```
['Sachin', 'GOAT', 'previous', 'generation', '.', 'Virat', 'GOAT',
'generation', '.', 'Shubman', 'GOAT', 'next', 'generation']
```

```
#Stemming

from nltk.stem import PorterStemmer

stemmer = PorterStemmer()

stemmed_tokens = []
for token in cleaned_tokens:
  stemmed = stemmer.stem(token)
  stemmed_tokens.append(stemmed)
print(stemmed_tokens)
```

```
['sachin', 'goat', 'previou', 'gener', '.', 'virat', 'goat', 'gener',
'.', 'shubman', 'goat', 'next', 'gener']
```

```
#Lemmatization

from nltk.stem import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()

lemmatized_tokens = []
for token in cleaned_tokens:
  lemmatized = lemmatizer.lemmatize(token)
  lemmatized_tokens.append(lemmatized)
print(lemmatized_tokens)
```

```
['Sachin', 'GOAT', 'previous', 'generation', '.', 'Virat', 'GOAT',
'generation', '.', 'Shubman', 'GOAT', 'next', 'generation']
```

```
#TF-IDF

from sklearn.feature_extraction.text import TfidfVectorizer

corpus = [
    "Sachin was the GOAT of the previous generation",
    "Virat is the GOAT of the this generation",
    "Shubman will be the GOAT of the next generation"
]

vectorizer = TfidfVectorizer()

matrix = vectorizer.fit(corpus)
matrix.vocabulary_
```

```
{'sachin': 7,
 'was': 12,
 'the': 9,
 'goat': 2,
 'of': 5,
 'previous': 6,
 'generation': 1,
 'virat': 11,
 'is': 3,
 'this': 10,
 'shubman': 8,
 'will': 13,
 'be': 0,
 'next': 4}

tfidf_matrix = vectorizer.transform(corpus)
print(tfidf_matrix)

  (0, 12)  0.4286758743128819
  (0, 9)   0.5063657539459899
  (0, 7)   0.4286758743128819
  (0, 6)   0.4286758743128819
  (0, 5)   0.25318287697299496
  (0, 2)   0.25318287697299496
  (0, 1)   0.25318287697299496
  (1, 11)  0.4286758743128819
  (1, 10)  0.4286758743128819
  (1, 9)   0.5063657539459899
  (1, 5)   0.25318287697299496
  (1, 3)   0.4286758743128819
  (1, 2)   0.25318287697299496
  (1, 1)   0.25318287697299496
  (2, 13)  0.39400039808922477
  (2, 9)   0.4654059642457353
  (2, 8)   0.39400039808922477
  (2, 5)   0.23270298212286766
  (2, 4)   0.39400039808922477
  (2, 2)   0.23270298212286766
  (2, 1)   0.23270298212286766
  (2, 0)   0.39400039808922477

print(vectorizer.get_feature_names_out())

['be' 'generation' 'goat' 'is' 'next' 'of' 'previous' 'sachin'
'shubman'
 'the' 'this' 'virat' 'was' 'will']
```