

```
import nltk
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('averaged_perceptron_tagger')
from nltk import word_tokenize, sent_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer, WordNetLemmatizer
from nltk import pos_tag
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
import pandas as pd
```

```
sent = "Sachin is considered to be one of the greatest cricket players. Virat is the captain of the Indian cricket team."
```

```
print("----- SENTENCES -----")
```

```
print(sent)
```

```
print("----- WORD TOKENIZATION -----")
```

```
print(word_tokenize(sent))
```

```
print("----- SENT TOKENIZATION -----")
```

```
print(sent_tokenize(sent))
```

```
stop_words = set(stopwords.words('english'))
```

```
token = word_tokenize(sent)
```

```
cleaned_token = [word for word in token if word.lower() not in stop_words]
```

```
words = [cleaned_word.lower() for cleaned_word in cleaned_token if cleaned_word.isalpha()]
```

```
print("----- WORDS -----")
```

```
print(words)
```

```
# STEMMING
```

```
stemmer = PorterStemmer()
```

```
port_stemmer_output = [stemmer.stem(word) for word in words]
print("----- STEMMING -----")
print(port_stemmer_output)
```

```
# LEMMATIZATION
```

```
lemmatizer = WordNetLemmatizer()
lemmatizer_output = [lemmatizer.lemmatize(word) for word in words]
print("----- LEMMATIZATION -----")
print(lemmatizer_output)
```

```
# POS TAGGING
```

```
tagged = pos_tag(cleaned_token)
print("----- POS TAGGING -----")
print(tagged)
```

```
docs = [
    "Sachin is considered to be one of the greatest cricket players.",
    "Federer is considered one of the greatest tennis players.",
    "Nadal is considered one of the greatest tennis players.",
    "Virat is the captain of the Indian cricket team."
]
```

```
vectorizer = TfidfVectorizer(analyzer="word", norm=None, use_idf=True, smooth_idf=True)
tfidfMat = vectorizer.fit_transform(docs)
```

```
features_names = vectorizer.get_feature_names_out()
print("----- FEATURE NAMES -----")
print(features_names)
```

```
dense = tfidfMat.todense()
denselist = dense.tolist()
df = pd.DataFrame(denselist, columns=features_names)
```

```
docsList = ['Docs_1', 'Docs_2', 'Docs_3', 'Docs_4']

skDocsIflfdhf = pd.DataFrame(tfidfMat.todense(), index=docsList, columns=features_names)

print("----- SK DOCS -----")

print(skDocsIflfdhf)
```

```
csim = cosine_similarity(tfidfMat, tfidfMat)
```

```
csimDf = pd.DataFrame(csim, index=docsList, columns=docsList)
```

```
print("----- COSINE SIMILARITY -----")
```

```
print(csimDf)
```

OUTPUT-

```

----- SENTENCES -----
Sachin is considered to be one of the greatest cricket players. Virat is the captain of the Indian cricket team.
----- WORD TOKENIZATION -----
['Sachin', 'is', 'considered', 'to', 'be', 'one', 'of', 'the', 'greatest', 'cricket', 'players', '.', 'Virat', 'is', 'the', 'captain', 'of', 'the', 'Indian', 'cricket', 'team', '.']
----- SENT TOKENIZATION -----
['Sachin is considered to be one of the greatest cricket players.', 'Virat is the captain of the Indian cricket team.']
----- WORDS -----
['sachin', 'considered', 'one', 'greatest', 'cricket', 'players', 'virat', 'captain', 'indian', 'cricket', 'team']
----- STEMMING -----
['sachin', 'consid', 'one', 'greatest', 'cricket', 'player', 'virat', 'captain', 'indian', 'cricket', 'team']
----- LEMMATIZATION -----
['sachin', 'considered', 'one', 'greatest', 'cricket', 'player', 'virat', 'captain', 'indian', 'cricket', 'team']
----- POS TAGGING -----
[('Sachin', 'NNP'), ('considered', 'VBD'), ('one', 'CD'), ('greatest', 'JJ'), ('cricket', 'NN'), ('players', 'NNS'), ('.', '.'), ('Virat', 'NNP'), ('captain', 'NN'), ('Indian', 'JJ'), ('cricket', 'NN'), ('team', 'NN'), ('.', '.')]
----- FEATURE NAMES -----
['be', 'captain', 'considered', 'cricket', 'federer', 'greatest', 'indian', 'is', 'nadal', 'of', 'one', 'players', 'sachin', 'team', 'tennis', 'the', 'to', 'virat']
----- SK DOCS -----
be captain considered cricket federer greatest ... sachin team tennis the to virat
Docs_1 1.916291 0.000000 1.223144 1.510826 0.000000 1.223144 ... 1.916291 0.000000 0.000000 1.0 1.916291 0.000000
Docs_2 0.000000 0.000000 1.223144 0.000000 1.916291 1.223144 ... 0.000000 0.000000 1.510826 1.0 0.000000 0.000000
Docs_3 0.000000 0.000000 1.223144 0.000000 0.000000 1.223144 ... 0.000000 0.000000 1.510826 1.0 0.000000 0.000000
Docs_4 0.000000 1.916291 0.000000 1.510826 0.000000 0.000000 ... 0.000000 1.916291 0.000000 2.0 0.000000 1.916291

[4 rows x 18 columns]
----- COSINE SIMILARITY -----
Docs_1 Docs_2 Docs_3 Docs_4
Docs_1 1.000000 0.492416 0.492416 0.277687
Docs_2 0.492416 1.000000 0.754190 0.215926
Docs_3 0.492416 0.754190 1.000000 0.215926
Docs_4 0.277687 0.215926 0.215926 1.000000

```