

```
import nltk
import string
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer, WordNetLemmatizer
from nltk import word_tokenize, pos_tag
from sklearn.feature_extraction.text import TfidfVectorizer
```

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

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

```
# Sample text
text = "NLP is useful and important for AI."
```

```
# 1. Tokenization
tokens = word_tokenize(text)
```

```
# 2. POS Tagging
pos_tags = pos_tag(tokens)
```

```
# 3. Stop words removal
stop_words = set(stopwords.words('english'))
filtered = [w for w in tokens if w.lower() not in stop_words and w not in string.punctuation]
```

```
# 4. Stemming
stemmer = PorterStemmer()
stemmed = [stemmer.stem(w) for w in filtered]
```

```
# 5. Lemmatization
lemmatizer = WordNetLemmatizer()
lemmatized = [lemmatizer.lemmatize(w) for w in filtered]
```

```
print("Tokens:", tokens)
print("POS Tags:", pos_tags)
print("After Stopword Removal:", filtered)
print("Stemming:", stemmed)
print("Lemmatization:", lemmatized)
```

```
Tokens: ['NLP', 'is', 'useful', 'and', 'important', 'for', 'AI', '.']
POS Tags: [('NLP', 'NNP'), ('is', 'VBZ'), ('useful', 'JJ'), ('and', 'CC'), ('important', 'JJ'), ('for', 'IN'), ('AI', 'NNP'), ('.', '.')]
After Stopword Removal: ['NLP', 'useful', 'important', 'AI']
Stemming: ['nlp', 'use', 'import', 'ai']
Lemmatization: ['NLP', 'useful', 'important', 'AI']
```

```
# 6. TF-IDF Representation
docs = ["Natural Language Processing is cool", "Language models are powerful in AI"]
vectorizer = TfidfVectorizer()
tfidf_matrix = vectorizer.fit_transform(docs)
```

```
print("\nTF-IDF Feature Names:", vectorizer.get_feature_names_out())
print("TF-IDF Matrix:\n", tfidf_matrix.toarray())
```

```
TF-IDF Feature Names: ['ai' 'are' 'cool' 'in' 'is' 'language' 'models' 'natural' 'powerful'
'processing']
TF-IDF Matrix:
[[0.         0.         0.47107781 0.         0.47107781 0.33517574
 0.         0.47107781 0.         0.47107781]
 [0.4261596 0.4261596 0.         0.4261596 0.         0.30321606
 0.4261596 0.         0.4261596 0.         ]]
```

Start coding or [generate](#) with AI.