### Importing Necessary Libraries

```
import nltk
import string
import numpy as np
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer, WordNetLemmatizer
from sklearn.feature_extraction.text import TfidfVectorizer
```

### Download necessary resources

```
nltk.download('averaged_perceptron_tagger_eng')
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('punkt_tab')
```

```
[nltk_data] Downloading package averaged_perceptron_tagger_eng to
[nltk_data]     /root/nltk_data...
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger_eng.zip.
[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!
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Package punkt_tab is already up-to-date!
True
```

### Sample document

```
document = """Text analytics is a field of study that uses natural language processing (NLP)
and machine learning techniques to analyze text data. It helps in extracting meaningful
insights from unstructured text."""
```

### Tokenization

```
tokens = word_tokenize(document)
print("Tokenized Words:", tokens)
```

```
Tokenized Words: ['Text', 'analytics', 'is', 'a', 'field', 'of', 'study', 'that', 'uses', 'natural'
```

### POS Tagging

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

```
POS Tags: [('Text', 'NN'), ('analytics', 'NNS'), ('is', 'VBZ'), ('a', 'DT'), ('field', 'NN'), ('of'
```

### Stop Words Removal

```
stop_words = set(stopwords.words('english'))
filtered_tokens = [word for word in tokens if word.lower() not in stop_words and word not in string.pun
```

```
print("\nTokens after Stop Words Removal:", filtered_tokens)
```

```
Tokens after Stop Words Removal: ['Text', 'analytics', 'field', 'study', 'uses', 'natural', 'langua
```

### Stemming

```
stemmer = PorterStemmer()
stemmed_words = [stemmer.stem(word) for word in filtered_tokens]
print("\nStemmed Words:", stemmed_words)
```

```
Stemmed Words: ['text', 'analyt', 'field', 'studi', 'use', 'natur', 'languag', 'process', 'nlp', 'm
```

### Lemmatization

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

```
Lemmatized Words: ['Text', 'analytics', 'field', 'study', 'us', 'natural', 'language', 'processing'
```

### Creating a list of documents for TF-IDF calculation

```
documents = [
    "Text analytics uses NLP and machine learning to analyze text.",
    "Natural language processing helps in extracting insights from text.",
    "Machine learning is useful for text classification and clustering."
]
```

### TF-IDF Calculation

```
vectorizer = TfidfVectorizer()
tfidf_matrix = vectorizer.fit_transform(documents)
```

### Convert to array and display

```
tfidf_array = tfidf_matrix.toarray()
print("\nTF-IDF Matrix:\n", tfidf_array)
```

```
TF-IDF Matrix:
 [[0.35070436 0.35070436 0.2667197  0.         0.         0.
  0.         0.         0.         0.         0.         0.
  0.         0.2667197  0.2667197  0.         0.35070436 0.
  0.41426329 0.35070436 0.         0.35070436]
 [0.         0.         0.         0.         0.         0.34608857
  0.         0.34608857 0.34608857 0.34608857 0.34608857 0.
  0.34608857 0.         0.         0.34608857 0.         0.34608857
  0.20440549 0.         0.         0.        ]
 [0.         0.         0.28574186 0.37571621 0.37571621 0.
  0.37571621 0.         0.         0.         0.         0.37571621
  0.         0.28574186 0.28574186 0.         0.         0.
  0.22190405 0.         0.37571621 0.        ]]
```

**Display Feature Names**

```
print("\nFeature Names (Words in TF-IDF):", vectorizer.get_feature_names_out())
```

```
Feature Names (Words in TF-IDF): ['analytics' 'analyze' 'and' 'classification' 'clustering' 'extrac
 'for' 'from' 'helps' 'in' 'insights' 'is' 'language' 'learning' 'machine'
 'natural' 'nlp' 'processing' 'text' 'to' 'useful' 'uses']
```

**Conclusion**

The document used in this study is a sample text related to text analytics and NLP. It undergoes preprocessing using tokenization, POS tagging, stop words removal, stemming, and lemmatization. These preprocessing techniques help in cleaning and structuring textual data for further analysis. TF-IDF representation is used to transform text into numerical form, helping in feature extraction, text classification, and clustering. The heatmap visualization of the TF-IDF matrix highlights the significance of different terms across multiple documents. Overall, these techniques are fundamental for various NLP applications, such as search engines, sentiment analysis, and document similarity analysis.