practical-7-dsbda-1

May 4, 2025

[5]:

```
#Practical 7
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer, WordNetLemmatizer
from nltk import pos_tag
from sklearn.feature_extraction.text import TfidfVectorizer
import string
```

[6]:

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

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

[6]: True

[7]:

```
sample_doc = "This is a practical of dsbda"
```

[9]:

```
tokens = word_tokenize(sample_doc)
print("Tokens:",tokens)
```

```
Tokens: ['This', 'is', 'a', 'practical', 'of', 'dsbda']
```

[10]:

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

```
POS Tags: [('This', 'DT'), ('is', 'VBZ'), ('a', 'DT'), ('practical', 'JJ'),
('of', 'IN'), ('dsbda', 'NN')]
```

[11]:

```
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.punctuation]
print("After Stop Word Removal:",filtered_tokens)
```

```
After Stop Word Removal: ['practical', 'dsbda']
```

[12]:

```
stemmer = PorterStemmer()
stemmed = [stemmer.stem(word) for word in filtered_tokens]
print("Stemmed Words : ",stemmed)
```

```
Stemmed Words :  ['practic', 'dsbda']
```

[13]:

```
lemmatizer = WordNetLemmatizer()
lemmatized = [lemmatizer.lemmatize(word) for word in filtered_tokens]
print("Lemmatized Words:",lemmatized)
```

```
Lemmatized Words: ['practical', 'dsbda']
```

[16]:

```
import pandas as pd
documents = ["This the practical of dsbda"]

tfidf_vectorizer = TfidfVectorizer()
tfidf_matrix = tfidf_vectorizer.fit_transform(documents)

df_tfidf = pd.DataFrame(tfidf_matrix.toarray(), columns=tfidf_vectorizer.
  ↪get_feature_names_out())
print("TF-IDF Representation:\n",df_tfidf)
```

```
TF-IDF Representation:
       dsbda        of  practical       the      this
0  0.447214  0.447214   0.447214  0.447214  0.447214
```

[ ]:

```
```