Assignment -2

Name:POTHU

RAHUL

Ht.no: 2203A51439

Batch: 12

Course name: NLP (Natural Language Processing)

1. Take your own text or take text as “Hello there! How are you doing today? NLP is fascinating." Implement Tokenization in the text.

```
import nltk
nltk.download('punkt')
from nltk.tokenize import sent_tokenize, word_tokenize

text = """Hello there! How are you doing today?
NLP is fascinating."""

def paragraph_tokenize(text):
    paragraphs = text.split('\n\n')
    return paragraphs

paragraphs = paragraph_tokenize(text)
tokenized_text = []

for paragraph in paragraphs:
    sentences = sent_tokenize(paragraph)
    tokenized_sentences = [word_tokenize(sentence) for sentence in sentences]
    tokenized_text.append(tokenized_sentences)

tokenized_text
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\HP\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[[['Hello', 'there', '!'],
  ['How', 'are', 'you', 'doing', 'today', '?'],
  ['NLP', 'is', 'fascinating', '.']]]
```

2. Take your own words or take words = ["running", "ran", "runs", "easily", "fairly"]. Implement Stemming in the text.

```python
import nltk
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize
import pandas as pd

data = {
    'Text': ["running", "ran", "runs", "easily", "fairly"]
}

df = pd.DataFrame(data)
ps = PorterStemmer()

def stem_words(text):
    return ps.stem(text)

df['Stemmed_Text'] = df['Text'].apply(stem_words)
print(df[['Text', 'Stemmed_Text']])

stemmed_file_path = "stemmed_text_dataset.csv"
df.to_csv(stemmed_file_path, index=False)
```

```
      Text Stemmed_Text
0  running          run
1      ran          ran
2     runs          run
3   easily       easili
4   fairly       fairli
```

	Text	Stemmed_Text
1	running	run
2	ran	ran
3	runs	run
4	easily	easili
5	fairly	fairli

3. Implement representation of word on any text or take text as “NLP is fun and interesting.", "NLP involves linguistics and computer science."

```
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer

text = ["NLP is fun and interesting.",
        "NLP involves linguistics and computer science."]

count_vect = CountVectorizer()
count_vector = count_vect.fit_transform(text)
print("Count Vector Representation:\n", count_vector.toarray())
print("Feature Names:", count_vect.get_feature_names_out())

tfidf_vect = TfidfVectorizer()
tfidf_vector = tfidf_vect.fit_transform(text)
print("\nTF-IDF Vector Representation:\n", tfidf_vector.toarray())
print("Feature Names:", tfidf_vect.get_feature_names_out())
```

```
Count Vector Representation:
 [[1 0 1 1 0 1 0 1 0]
 [1 1 0 0 1 0 1 1 1]]
Feature Names: ['and' 'computer' 'fun' 'interesting' 'involves' 'is' 'linguistics' 'nlp'
 'science']

TF-IDF Vector Representation:
 [[0.35520009 0.         0.49922133 0.49922133 0.         0.49922133
  0.         0.35520009 0.        ]
 [0.31779954 0.44665616 0.         0.         0.44665616 0.
  0.44665616 0.31779954 0.44665616]]
Feature Names: ['and' 'computer' 'fun' 'interesting' 'involves' 'is' 'linguistics' 'nlp'
 'science']
```

4. Implement Representation of Sentences on following or take any other sentence,"NLP is an interesting field.", "It involves processing natural language."

```python
from sklearn.feature_extraction.text import CountVectorizer
import pandas as pd

sentences = ["NLP is an interesting field.", "It involves processing natural language."]
vectorizer = CountVectorizer()
bow_matrix = vectorizer.fit_transform(sentences)
bow_df = pd.DataFrame(bow_matrix.toarray(), columns=vectorizer.get_feature_names_out())

print("Bag of Words Representation:")
print(bow_df)
```

```
Bag of Words Representation:
   an  field  interesting  involves  is  it  language  natural  nlp  \
0   1      1            1         0   1   0         0        0    1
1   0      0            0         1   0   1         1        1    0

   processing
0           0
1           1
```