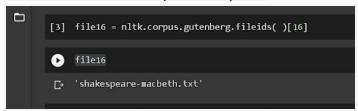
Name:Puja Bathija SUID:692880674

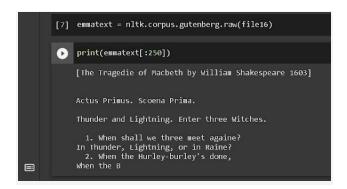
Title: Natural Language Processing lab installing nltk

Book title: Macbeth by Shakespeare



## Steps:

1. Get the text with nltk.corpus.gutenberg.raw() raw() function is used to convert the information into a string[1].



2. Get the tokens with nltk.word\_tokenize()
This is the next step after getting the string to divide it into words this is done using the word\_tokenize() function[2].

```
[9] nltk.download('punkt')
    emmatokens = nltk.word_tokenize(emmatext)

[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Unzipping tokenizers/punkt.zip.

[24] print(emmatokens)

['[', 'The', 'Tragedie', 'of', 'Macbeth', 'by', 'William', 'Shakespeare', '1603', ']', 'Actus', 'Primus', '.', 'Scottant'
]
```

3. Get the words by using w.lower() to lowercase the tokens

The words are converted into the lower case as a measure to prevent the same words but in a different form to be identified as different words. For example, Apple and apple would be identified as different words but once all words are converted into the lower case the text would be uniform[1].

- 4. Make the frequency distribution with FreqDist
  The FreqDist is a function in nltk library that takes the words and returns words and their frequency[3].
- Get the 30 top frequency words with most\_common(30) and print the word, frequency pairs.
   most\_common() function returns the most frequent words and their count[3].

List of the top 30 frequent words

```
C (',', 1962)
('.', 1219)
      ('thé', 649)
       'and', 545)
       ':', 477)
'to', 383)
      ('of', 338)
       'i', 331)
       '?', 241)
       'a', 239)
       'that', 236)
       'is', 211)
       'my', 203)
'you', 203)
       'in', 200)
       "'d", 192)
'not', 185)
       'it', 161)
       'with', 153)
       'his', 146)
       'be', 137)
'macb', 137)
       "'s", 128)
       'your', 126)
'our', 123)
'haue', 122)
       'but', 120)
'what', 117)
     ('me', 113)
('he', 112)
```

Observations:

Most of the top frequent words are either punctuations, pronounces, prepositions, and other filler words.

## Lessons learned:

- 1. To convert the data in a string using raw() function.
- 2. To break a string into words using word\_tokenize() function
- 3. To make a frequency distribution of the words using FreqDist and checking the most common words in the distribution using most\_common() function.