Experiment 03 NLP DLOC Harijan Ritik CSE(DS)

Library required

```
!pip install nltk
```

```
Requirement already satisfied: nltk in c:\users\admin\appdata\local\programs\python\python37\lib\site-packages (3.6.2)
Requirement already satisfied: joblib in c:\users\admin\appdata\local\programs\python\python37\lib\site-packages (from nltk) (1.0.0
Requirement already satisfied: click in c:\users\admin\appdata\local\programs\python\python37\lib\site-packages (from nltk) (7.1.2)
Requirement already satisfied: regex in c:\users\admin\appdata\local\programs\python\python37\lib\site-packages (from nltk) (2021.4
Requirement already satisfied: tqdm in c:\users\admin\appdata\local\programs\python\python37\lib\site-packages (from nltk) (4.60.0)
WARNING: You are using pip version 22.0; however, version 23.2.1 is available.
You should consider upgrading via the 'c:\users\admin\appdata\local\programs\python\python37\python.exe -m pip install --upgrade pi
```

Text

```
text = 'TON 618 is a hyperluminous, broad-absorption-line, radio-loud quasar and Lyman-alpha blob located near the border of the constell
```

```
text
```

```
'TON 618 is a hyperluminous, broad-absorption-line, radio-loud quasar and Lyman-alpha blob located near the border of the
constellations Canes Venatici and Coma Berenices, with the projected comoving distance of approximately 18.2 billion light-years
from Earth.'
```

Stopwords

```
from nltk.corpus import stopwords
```

```
stop_words = stopwords.words('english')
```

```
from nltk.tokenize import word_tokenize
words = word_tokenize(text)
```

Applying stop words

```
holder = list()
for w in words:
    if w not in set(stop_words):
        holder.append(w)
```

```
holder
```

```
['TON',
 '618',
 'hyperluminous',
 ',',
 'broad-absorption-line',
 ',',
 'radio-loud',
 'quasar',
 'Lyman-alpha',
 'blob',
 'located',
 'near',
 'border',
 'constellations',
 'Canes',
 'Venatici',
 'Coma',
 'Berenices',
 ',',
 'projected',
 'comoving',
 'distance',
 'approximately',
 '18.2',
 'billion',
 'light-years',
 'Earth',
 '.']
```

List Comprehension for stop words

```
holder = [w for w in words if w not in set(stop_words)]
print(holder)
```

```
['TON', '618', 'hyperluminous', ',', 'broad-absorption-line', ',', 'radio-loud', 'quasar', 'Lyman-alpha', 'blob', 'located', 'near'
```

Stemming

```
from nltk.stem import PorterStemmer, SnowballStemmer, LancasterStemmer
```

```
porter = PorterStemmer()
snow = SnowballStemmer(language = 'english')
lancaster = LancasterStemmer()
```

```
words = ['play', 'plays', 'played', 'playing', 'player']
```

Porter Stemmer

```
porter_stemmed = list()
for w in words:
    stemmed_words = porter.stem(w)
    porter_stemmed.append(stemmed_words)
```

```
porter_stemmed
```

```
['play', 'play', 'play', 'play', 'player']
```

Porter Stemmer List Comprehension

```
porter_stemmed = [porter.stem(x) for x in words]
print (porter_stemmed)
```

```
['play', 'play', 'play', 'play', 'player']
```

Snowball Stemmer

```
snow_stemmed = list()
for w in words:
    stemmed_words = snow.stem(w)
    snow_stemmed.append(stemmed_words)
```

```
snow_stemmed
```

```
['play', 'play', 'play', 'play', 'player']
```

Snowball Stemmer List Comprehension

```
snow_stemmed = [snow.stem(x) for x in words]
print (snow_stemmed)
```

```
['play', 'play', 'play', 'play', 'player']
```

Lancaster Stemmer

```
lancaster_stemmed = list()
for w in words:
    stemmed_words = lancaster.stem(w)
    lancaster_stemmed.append(stemmed_words)
```

```
lancaster_stemmed
```

```
['play', 'play', 'play', 'play', 'play']
```

Lancaster Stemmer List Comprehension

```
lancaster_stemmed = [lancaster.stem(x) for x in words]
print (lancaster_stemmed)
```

```
['play', 'play', 'play', 'play', 'play']
```

Lemmatization : This has a more expansive vocabulary than Stemming

```
from nltk.stem import WordNetLemmatizer
wordnet = WordNetLemmatizer()
```

```
lemmatized = [wordnet.lemmatize(x) for x in words]
```

```
lemmatized
```

```
['play', 'play', 'played', 'playing', 'player']
```