```
###Pratical 7 - Text Analytics
1. Extract Sample document and apply followir
Tokenization, POS Tagging, stop words removal
Create representation of document by calculat
Document Frequency
```

Pratical 7 - Text Analytics

1. Extract Sample document and apply following document preprocessing methods: Tokenization, POS Tagging, stop words removal, Stemming and Lemmatization. Create representation of document by calculating Term Frequency and Inverse Document Frequency

```
import pandas as pd
import numpy as np
```

```
text = '''It was a Thursday, but it felt like a Monday to John. And John loved Mondays. He

I should probably get another latte. I've just been sitting here with this empty cup. But

John was always impatient on the weekends; he missed the formal structure of the business

Jesus, I've written another loser. '''
```

Tokenization of text

```
text_split = text.split()
```

```
text
```

```
'It was a Thursday, but it felt like a Monday to John. And John loved Mondays. He th
rived at work. He dismissed the old cliché of dreading Monday mornings and refused t
o engage in water-cooler complaints about "the grind" and empty conversations that i
ncluded the familiar parry "How was your weekend?" "Too short!". Yes, John liked his
work and was unashamed.\n\nI should probably get another latte. I've just been sitti
ng here with this empty cup. But then I'll start to get jittery. I'll get a decaf. N
```

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

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package averaged_perceptron_tagger to
```

```
[nltk_data]     /root/nltk_data...
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger.zip.
True
```

```
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize, sent_tokenize
stop_words = stopwords.words('english')


tokenized = sent_tokenize(text)
for i in tokenized:

    # Word tokenizers is used to find the words
    # and punctuation in a string
    wordsList = nltk.word_tokenize(i)

    # removing stop words from wordList
    wordsList = [w for w in wordsList if not w in stop_words]

    #  Using a Tagger. Which is part-of-speech
    # tagger or POS-tagger.
    tagged = nltk.pos_tag(wordsList)

    print(tagged)
```

```
[('It', 'PRP'), ('Thursday', 'NNP'), (',', ','), ('felt', 'VBD'), ('like', 'IN'), ('M
[('And', 'CC'), ('John', 'NNP'), ('loved', 'VBD'), ('Mondays', 'NNP'), ('.', '.')]
[('He', 'PRP'), ('thrived', 'VBD'), ('work', 'NN'), ('.', '.')]
[('He', 'PRP'), ('dismissed', 'VBD'), ('old', 'JJ'), ('cliché', 'NN'), ('dreading',
[('Yes', 'UH'), (',', ','), ('John', 'NNP'), ('liked', 'VBD'), ('work', 'NN'), ('unas
[('I', 'PRP'), ('probably', 'RB'), ('get', 'VB'), ('another', 'DT'), ('latte', 'NN'),
[('I', 'PRP'), ('’', 'VBP'), ('sitting', 'VBG'), ('empty', 'JJ'), ('cup', 'NN'), ('.
[('But', 'CC'), ('I', 'PRP'), ('’', 'VBP'), ('start', 'JJ'), ('get', 'VB'), ('jittery
[('I', 'PRP'), ('’', 'VBP'), ('get', 'VB'), ('decaf', 'NN'), ('.', '.')]
[('No', 'DT'), (',', ','), ('’', 'FW'), ('stupid', 'JJ'), (',', ','), ('feels', 'JJ')
[('I', 'PRP'), ('’', 'VBP'), ('justify', 'NN'), ('.', '.')]
[('John', 'NNP'), ('always', 'RB'), ('impatient', 'JJ'), ('weekends', 'NNS'), (';',
[('When', 'WRB'), ('younger', 'JJR'), ('used', 'VBD'), ('stay', 'NN'), ('late', 'JJ')
```

```
stopwords
```

```
<WordListCorpusReader in '/root/nltk_data/corpora/stopwords'>
```

```
print(stopwords)
```

```
<WordListCorpusReader in '/root/nltk_data/corpora/stopwords'>
```

Stemming and Lemmatization

1. Stemming

```
from nltk.stem.porter import PorterStemmer


porter_stemmer = PorterStemmer()


nltk_token = nltk.word_tokenize(text)


for w in nltk_token:
  print("Actual : %s , Stem: %s" %(w, porter_stemmer.stem(w)))
```

```
Actual : weekends , Stem: weekend
Actual : ; , Stem: ;
Actual : he , Stem: he
Actual : missed , Stem: miss
Actual : the , Stem: the
Actual : formal , Stem: formal
Actual : structure , Stem: structur
Actual : of , Stem: of
Actual : the , Stem: the
Actual : business , Stem: busi
Actual : week , Stem: week
Actual : . , Stem: .
Actual : When , Stem: when
Actual : he , Stem: he
Actual : was , Stem: wa
Actual : younger , Stem: younger
Actual : he , Stem: he
Actual : used , Stem: use
Actual : to , Stem: to
Actual : stay , Stem: stay
Actual : late , Stem: late
Actual : after , Stem: after
Actual : school , Stem: school
Actual : on , Stem: on
Actual : Fridays , Stem: friday
Actual : and , Stem: and
Actual : come , Stem: come
Actual : in , Stem: in
Actual : early , Stem: earli
Actual : on , Stem: on
Actual : Mondays , Stem: monday
Actual : , , Stem: ,
Actual : a , Stem: a
Actual : pattern , Stem: pattern
Actual : his , Stem: hi
Actual : mother , Stem: mother
Actual : referred , Stem: refer
Actual : to , Stem: to
Actual : with , Stem: with
Actual : equal , Stem: equal
Actual : parts , Stem: part
Actual : admiration , Stem: admir
Actual : and , Stem: and
Actual : disdain , Stem: disdain
Actual : as , Stem: as
Actual : “ , Stem: “
Actual : studying , Stem: studi
```

```
Actual : overtime. , Stem: overtime.
Actual : ” , Stem: ”
Actual : Jesus , Stem: jesu
Actual : , , Stem: ,
Actual : I , Stem: I
Actual : ’ , Stem: ’
Actual : ve , Stem: ve
Actual : written , Stem: written
Actual : another , Stem: anoth
Actual : loser , Stem: loser
Actual : . , Stem: .
```

2.Lemmatization

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

```
nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Unzipping corpora/wordnet.zip.
True
```

```
for w in nltk_token:
  print("Actual : %s , Lemme: %s" %(w, wordnet_lemmatizer.lemmatize(w)))
```

```
Actual : weekends , Lemme: weekend
Actual : ; , Lemme: ;
Actual : he , Lemme: he
Actual : missed , Lemme: missed
Actual : the , Lemme: the
Actual : formal , Lemme: formal
Actual : structure , Lemme: structure
Actual : of , Lemme: of
Actual : the , Lemme: the
Actual : business , Lemme: business
Actual : week , Lemme: week
Actual : . , Lemme: .
Actual : When , Lemme: When
Actual : he , Lemme: he
Actual : was , Lemme: wa
Actual : younger , Lemme: younger
Actual : he , Lemme: he
Actual : used , Lemme: used
Actual : to , Lemme: to
Actual : stay , Lemme: stay
Actual : late , Lemme: late
Actual : after , Lemme: after
Actual : school , Lemme: school
Actual : on , Lemme: on
Actual : Fridays , Lemme: Fridays
Actual : and , Lemme: and
Actual : come , Lemme: come
Actual : in , Lemme: in
Actual : early , Lemme: early
Actual : on , Lemme: on
Actual : Mondays , Lemme: Mondays
```

```
Actual : Mondays , Lemme: Mondays
Actual : , , Lemme: ,
Actual : a , Lemme: a
Actual : pattern , Lemme: pattern
Actual : his , Lemme: his
Actual : mother , Lemme: mother
Actual : referred , Lemme: referred
Actual : to , Lemme: to
Actual : with , Lemme: with
Actual : equal , Lemme: equal
Actual : parts , Lemme: part
Actual : admiration , Lemme: admiration
Actual : and , Lemme: and
Actual : disdain , Lemme: disdain
Actual : as , Lemme: a
Actual : “ , Lemme: “
Actual : studying , Lemme: studying
Actual : overtime. , Lemme: overtime.
Actual : ” , Lemme: ”
Actual : Jesus , Lemme: Jesus
Actual : , , Lemme: ,
Actual : I , Lemme: I
Actual : ’ , Lemme: ’
Actual : ve , Lemme: ve
Actual : written , Lemme: written
Actual : another , Lemme: another
Actual : loser , Lemme: loser
Actual : . , Lemme: .
```

✓ 2s completed at 9:55 AM