Tokenization

```
import nltk
nltk.download('punkt')
from nltk.tokenize import sent_tokenize,word_tokenize
data= "All work and no play makes jack a dull boy, all work and no play"
print(word_tokenize(data))
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
['All', 'work', 'and', 'no', 'play', 'makes', 'jack', 'a', 'dull', 'boy', ',', 'all', 'work', 'and', 'no', 'play']
```

Stop Word Removal

```
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
data= "All work and no play makes jack a dull boy, all work and no play makes jack a dull boy"
stopWords= set(stopwords.words('english'))
words= word_tokenize(data.lower())
wordsFiltered =[]
for w in words:
  if w not in stopWords:
    wordsFiltered.append(w)

print(wordsFiltered)
print(stopWords)
#print(stopWords.count())
```

```
['work', 'play', 'makes', 'jack', 'dull', 'boy', ',', 'work', 'play', 'makes', 'jack', 'dull', 'boy']
{'these', "aren't", 'yourself', 'my', 'o', 'be', 'up', 'does', 'wouldn', 'whom', 'myself', 'all', 'any', "wouldn't", 're
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Stemming

```
from nltk.stem import PorterStemmer
from nltk.tokenize import sent_tokenize,word_tokenize
words=["game","gaming","gamed","games"]
ps=PorterStemmer()
for word in words:
  print(ps.stem(word))
```

```
game
game
game
game
```

Regular Expression

```
path=r"C:\desktop\nayan"
print("raw string",path)
```

```
raw string C:\desktop\nayan
```

re.match()

```
import re
result = re.match('Analytics',r'Analytics Vidhya is the largest data science community of India')
print(result)
```

```
<re.Match object; span=(0, 9), match='Analytics'>
```

re.search()

```
result = re.search('founded',r'Andrew NG founded Coursera. He also founded deeplearning.ai')
print(result.group())
```

```
founded
```

re.findall()

```
result = re.findall('founded',r'Andrew NG founded Coursera. He also founded deeplearning.ai')
print(result)

    ['founded', 'founded']
```

Special Sequences

/b

```
str = r'Analytics Vidhya is the largest Analytics community of India'
x = re.findall(r"est\b", str)
print(x)

    ['est']
```

\d

```
str = "2 million monthly visits in Jan'19."

#Check if the string contains any digits (numbers from 0-9):

x = re.findall("\d", str)
print(x)

if (x):
   print("Yes, there is at least one match!")
else:
   print("No match")

    ['2', '1', '9']
    Yes, there is at least one match!
```

\D

```
str = "2 million monthly visits in Jan'19."

x = re.findall("\D", str)
print(x)

if (x):
  print("Yes, there is at least one match!")
else:
  print("No match")

    [' ', 'm', 'i', 'l', 'l', 'i', 'o', 'n', ' ', 'm', 'o', 'n', 't', 'h', 'l', 'y', ' ', 'v', 'i', 's', 'i', 't', 's', ' ',
    Yes, there is at least one match!
```

\w

```
str = "2 million monthly visits!"

x = re.findall("\w+",str)
print(x)

if (x):
  print("Yes, there is at least one match!")
else:
  print("No match")

    ['2', 'million', 'monthly', 'visits']
    Yes, there is at least one match!
```

\W

```
str = "2 million monthly visits9!"

x = re.findall("\W", str)
print(x)

if (x):
  print("Yes, there is at least one match!")
else:
  print("No match")
```

```
[' ', ' ', ' ', '!']
Yes, there is at least one match!
```