```
import nltk
```

```
nltk.download('stopwords')
nltk.download('words')
nltk.download('wordnet')
nltk.download('averged_perception_tagger')
nltk.download('punkt')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package words to /root/nltk_data...
[nltk_data]   Unzipping corpora/words.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data] Error loading averged_perception_tagger: Package
[nltk_data]     'averged_perception_tagger' not found in index
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
True
```

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

```
sent= "They told that thier eges are 20 23 and 27 respectively"
```

```
add=[]
```

```
for word in sent.split():
  if word.isdigit():
    add.append(int(word))
```

```
print ("Ave", sum(add)/len(add))
```

```
Ave 23.333333333333332
```

```
from nltk.tokenize import word_tokenize, sent_tokenize
```

```
sent= "Hello all! how are you? Welcome to pun "
```

```
sent_tokenize(sent)
```

```
['Hello\xa0all!\xa0how\xa0are\xa0you?', 'Welcome\xa0to\xa0pun']
```

```
word_tokenize(sent)
```

```
['Hello', 'all', '!', 'how', 'are', 'you', '?', 'Welcome', 'to', 'pun']
```

```
from nltk.tokenize import SpaceTokenizer
tk=SpaceTokenizer()
tk.tokenize(sent)
```

```
['Hello\xa0all!\xa0how\xa0are\xa0you?\xa0Welcome\xa0to\xa0pun', '']
```

```
sent='Hello all!\tHow are u?\tto pune'
```

```
print(sent)
```

```
Hello all!	How are u?	to pune
```

```
s1='ctas','catlike','catty','cat'
s2='stemmer','stemming','stemmed','stem'
s3='fishing','fished','fisher','fish'
s4='argue','argued','argues','argus'
```

```
from nltk.stem import PorterStemmer
```

```
ps=PorterStemmer()
```

```
ps.stem(s3[0])
```

```
'fish'
```

```
for word in s4:
  ps=PorterStemmer()
  print(ps.stem(word))
```

```
argu
argu
argu
argu
```

```
# lemmatization
```

```
word='playing'
```

```
from nltk.stem import WordNetLemmatizer
```

```
wnl=WordNetLemmatizer()
print(wnl.lemmatize(word,'n'))  # noun
print(wnl.lemmatize(word,'v'))  # verb
print(wnl.lemmatize(word,'a'))  # adjective
print(wnl.lemmatize(word,'r'))  # adverb
```

```
playing
play
playing
playing
```

```
word='went'
```

```
wnl=WordNetLemmatizer()
print(wnl.lemmatize(word,'n')) # noun
print(wnl.lemmatize(word,'v')) # verb
print(wnl.lemmatize(word,'a')) # adjective
print(wnl.lemmatize(word,'r')) # adverb
```

```
went
go
went
went
```

```
# POS tagging
```

```
from nltk import pos_tag
```

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

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

```
sents='Rajgad (literal meaning Ruling Fort) is a hill fort situated in the Pune district of Maharashtra, India. Formerly known as Murumde
```

```
print(sents)
```

```
Rajgad (literal meaning Ruling Fort) is a hill fort situated in the Pune district of Maharashtra, India. Formerly known as Murumdev
```

```
words=word_tokenize(sents)
```

```
nltk.download('omw-1.4')
```

```
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
True
```

```
pos_tag(words)
```

```
[('Rajgad', 'NNP'),
 ('(', '('),
 ('literal', 'JJ'),
 ('meaning', 'NN'),
 ('Ruling', 'NNP'),
 ('Fort', 'NNP'),
 (')', ')'),
 ('is', 'VBZ'),
```

```
 ('a', 'DT'),
 ('hill', 'NN'),
 ('fort', 'NN'),
 ('situated', 'VBN'),
 ('in', 'IN'),
 ('the', 'DT'),
 ('Pune', 'NNP'),
 ('district', 'NN'),
 ('of', 'IN'),
 ('Maharashtra', 'NNP'),
 (',', ','),
 ('India', 'NNP'),
 ('.', '.'),
 ('Formerly', 'RB'),
 ('known', 'VBN'),
 ('as', 'IN'),
 ('Murumdev', 'NNP'),
 (',', ','),
 ('the', 'DT'),
 ('fort', 'NN'),
 ('was', 'VBD'),
 ('the', 'DT'),
 ('capital', 'NN'),
 ('of', 'IN'),
 ('the', 'DT'),
 ('Maratha', 'NNP'),
 ('Empire', 'NNP'),
 ('under', 'IN'),
 ('the', 'DT'),
 ('rule', 'NN'),
 ('of', 'IN'),
 ('Shivaji', 'NNP'),
 ('for', 'IN'),
 ('almost', 'RB'),
 ('26', 'CD'),
 ('years', 'NNS'),
 (',', ','),
 ('afterwhich', 'IN'),
 ('the', 'DT'),
 ('capital', 'NN'),
 ('was', 'VBD'),
 ('moved', 'VBN'),
 ('to', 'TO'),
 ('the', 'DT'),
 ('Raigad', 'NNP'),
 ('Fort', 'NNP'),
 ('.', '.'),
 ('[', 'CC'),
 ('1', 'CD'),
 (']', 'NN'),
```

```
tags=pos_tag(words)
```

```
for word in tags:
  if word[1].startswith('V'):
    print(word[0])
```

```
 is
 situated
 known
 was
 was
 moved
 discovered
 called
 were
 used
 build
 fortify
 needed
```

```
# spell correction
from textblob import TextBlob
```

```
t=TextBlob('computoor')
print(t.correct())
```

```
 computer
```

```
t=TextBlob('nead')
print(t.correct())
```

```
 head
```