```
!pip install nltk
```

```
Defaulting to user installation because normal site-packages is not
writeable
Requirement already satisfied: nltk in c:\programdata\anaconda3\lib\
site-packages (3.8.1)
Requirement already satisfied: click in c:\programdata\anaconda3\lib\
site-packages (from nltk) (8.1.7)
Requirement already satisfied: joblib in c:\programdata\anaconda3\lib\
site-packages (from nltk) (1.4.2)
Requirement already satisfied: regex>=2021.8.3 in c:\programdata\
anaconda3\lib\site-packages (from nltk) (2023.10.3)
Requirement already satisfied: tqdm in c:\programdata\anaconda3\lib\
site-packages (from nltk) (4.66.4)
Requirement already satisfied: colorama in c:\programdata\anaconda3\
lib\site-packages (from click->nltk) (0.4.6)
```

```python
import nltk
nltk.download('punkt', download_dir='./nltk_data')
```

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

```
True
```

```python
#Give Input as ant test
text = "It is a truth universally acknowledged, that a single man in
possession of a good fortune,  must be in want of a wife."
text = text.lower()
print(text)
```

```
it is a truth universally acknowledged, that a single man in
possession of a good fortune,  must be in want of a wife.
```

```python
text = "It is a truth universally acknowledged, that a single man in
possession of a good fortune,  must be in want of a wife."
text = text.lower()
print(text)
```

```
it is a truth universally acknowledged, that a single man in
possession of a good fortune,  must be in want of a wife.
```

```python
import string
print(string.punctuation)
```

```
!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
```

```python
text_p = "".join([char for char in text if char not in
string.punctuation]); print(text_p)
```

```
it is a truth universally acknowledged that a single man in possession
of a good fortune  must be in want of a wife
```

```python
from nltk import word_tokenize, sent_tokenize

# Tokenize
words = word_tokenize(text_p)
words1 = sent_tokenize(text_p)
print(words)
print(words1)
```

```
['it', 'is', 'a', 'truth', 'universally', 'acknowledged', 'that', 'a',
'single', 'man', 'in', 'possession', 'of', 'a', 'good', 'fortune',
'must', 'be', 'in', 'want', 'of', 'a', 'wife']
['it is a truth universally acknowledged that a single man in
possession of a good fortune  must be in want of a wife']
```

```python
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\DELL\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

```
True
```

```python
nltk.download('stopwords')
from nltk.corpus import stopwords
stop_words = stopwords.words('english')
print(stop_words)
```

```
['a', 'about', 'above', 'after', 'again', 'against', 'ain', 'all',
'am', 'an', 'and', 'any', 'are', 'aren', "aren't", 'as', 'at', 'be',
'because', 'been', 'before', 'being', 'below', 'between', 'both',
'but', 'by', 'can', 'couldn', "couldn't", 'd', 'did', 'didn',
"didn't", 'do', 'does', 'doesn', "doesn't", 'doing', 'don', "don't",
'down', 'during', 'each', 'few', 'for', 'from', 'further', 'had',
'hadn', "hadn't", 'has', 'hasn', "hasn't", 'have', 'haven', "haven't",
'having', 'he', "he'd", "he'll", 'her', 'here', 'hers', 'herself',
"he's", 'him', 'himself', 'his', 'how', 'i', "i'd", 'if', "i'll",
"i'm", 'in', 'into', 'is', 'isn', "isn't", 'it', "it'd", "it'll",
"it's", 'its', 'itself', "i've", 'just', 'll', 'm', 'ma', 'me',
'mightn', "mightn't", 'more', 'most', 'mustn', "mustn't", 'my',
'myself', 'needn', "needn't", 'no', 'nor', 'not', 'now', 'o', 'of',
'off', 'on', 'once', 'only', 'or', 'other', 'our', 'ours',
'ourselves', 'out', 'over', 'own', 're', 's', 'same', 'shan',
"shan't", 'she', "she'd", "she'll", "she's", 'should', 'shouldn',
"shouldn't", "should've", 'so', 'some', 'such', 't', 'than', 'that',
"that'll", 'the', 'their', 'theirs', 'them', 'themselves', 'then',
'there', 'these', 'they', "they'd", "they'll", "they're", "they've",
'this', 'those', 'through', 'to', 'too', 'under', 'until', 'up', 've',
'very', 'was', 'wasn', "wasn't", 'we', "we'd", "we'll", "we're",
'were', 'weren', "weren't", "we've", 'what', 'when', 'where', 'which',
'while', 'who', 'whom', 'why', 'will', 'with', 'won', "won't",
'wouldn', "wouldn't", 'y', 'you', "you'd", "you'll", 'your', "you're",
'yours', 'yourself', 'yourselves', "you've"]
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\DELL\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

```python
filtered_words = [word for word in words if word not in stop_words]
print(filtered_words)
```

```
['truth', 'universally', 'acknowledged', 'single', 'man',
'possession', 'good', 'fortune', 'must', 'want', 'wife']
```

```python
from nltk.stem.porter import PorterStemmer
porter = PorterStemmer()
stemmed = [porter.stem(word) for word in filtered_words]
print(stemmed)
```

```
['truth', 'univers', 'acknowledg', 'singl', 'man', 'possess', 'good',
'fortun', 'must', 'want', 'wife']
```

```python
import nltk
nltk.download('averaged_perceptron_tagger')
from nltk import pos_tag
pos = pos_tag(filtered_words)
print(pos)
```

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\DELL\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping taggers\averaged_perceptron_tagger.zip.
```

```
[('truth', 'NN'), ('universally', 'RB'), ('acknowledged', 'VBD'),
('single', 'JJ'), ('man', 'NN'), ('possession', 'NN'), ('good', 'JJ'),
('fortune', 'NN'), ('must', 'MD'), ('want', 'VB'), ('wife', 'NN')]
```

```python
# 5. Calculate TF-IDF
from sklearn.feature_extraction.text import TfidfVectorizer
corpus = [text]
tfidf_vectorizer = TfidfVectorizer(stop_words='english')
tfidf_matrix = tfidf_vectorizer.fit_transform(corpus)

# Get feature names (terms)
terms = tfidf_vectorizer.get_feature_names_out()
print("TF-IDF Terms:", terms)
```

```
TF-IDF Terms: ['acknowledged' 'fortune' 'good' 'man' 'possession'
'single' 'truth'
 'universally' 'want' 'wife']
```

```python
# Display TF-IDF values for the document
tfidf_values = tfidf_matrix.toarray()
print("TF-IDF Values:", tfidf_values)
```

```
TF-IDF Values: [[0.31622777 0.31622777 0.31622777 0.31622777
0.31622777 0.31622777
  0.31622777 0.31622777 0.31622777 0.31622777]]
```