Vectorisation Technique : (stop words)

[1]:
```
import nltk
```

[ ]:
```
#download punkt model and punkt_tag mdel and stop words model
```

[3]:
```
nltk.download('punkt')
```

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

[3]:
```
True
```

[4]:
```
nltk.download('punkt_tab')
```

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

[4]:
```
True
```

[5]:
```
nltk.download('stop_words') #Wrtong Spelling of the packages.
```

```
[nltk_data] Error loading stop_words: Package 'stop_words' not found
[nltk_data]     in index
```

[5]:
```
False
```

[7]:
```
nltk.download('stopwords')
```

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

[7]:
```
True
```

[8]:
```
# import stopwords from the nltk.corpus
from nltk.corpus import stopwords
```

[9]:
```
# import the word tokenizer
from nltk.tokenize import word_tokenize
```

[11]:
```
corpus = "This is simple example of vectorisation and tokenisation using stops word."
```

[13]:
```
print('corpus:',corpus)
```

```
corpus: This is simple example of vectorisation and tokenisation using stops word.
```

[14]:
```
# we will perform tokenization
```

[15]:
```
words = word_tokenize(corpus)
```

[16]:
```
print('finding the words:',words)
```

```
finding the words: ['This', 'is', 'simple', 'example', 'of', 'vectorisation', 'and', 'tokenisation', 'using', 'stops', 'word', '.']
```

[17]:
```
# Removing the Un-necessary words:-
```

[19]:
```
english_stopwords = stopwords.words("english")
```

[20]:
```
print('English Words stop word:',english_stopwords)
```

```
English Words stop word: ['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"]
```

[22]:
```
filtered_words = []
for word in words:
    word = word.lower() # Lower word.
    if word in english_stopwords:
```

```
        pass
    else:
        filtered_words.append(word)
```

[23]:
```
print('Filtered Words without stop words:',filtered_words)
```

```
Filtered Words without stop words: ['simple', 'example', 'vectorisation', 'tokenisation', 'using', 'stops', 'word', '.']
```

[25]:
```
print('Length of Original Words :',len(words))
print('Length of Filtered Words :',len(filtered_words))
```

```
Length of Original Words : 12
Length of Filtered Words : 8
```