Saloni Bhingardive Roll No. 23 CSE (Data Science) NLP Practical No. 02

```
import nltk
nltk.download("all")
```

```
[nltk_data] Downloading collection 'all'
[nltk_data]    |
[nltk_data]    | Downloading package abc to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/abc.zip.
[nltk_data]    | Downloading package alpino to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/alpino.zip.
[nltk_data]    | Downloading package averaged_perceptron_tagger to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping taggers/averaged_perceptron_tagger.zip.
[nltk_data]    | Downloading package averaged_perceptron_tagger_eng to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping
[nltk_data]    |       taggers/averaged_perceptron_tagger_eng.zip.
[nltk_data]    | Downloading package averaged_perceptron_tagger_ru to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping
[nltk_data]    |       taggers/averaged_perceptron_tagger_ru.zip.
[nltk_data]    | Downloading package averaged_perceptron_tagger_rus to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping
[nltk_data]    |       taggers/averaged_perceptron_tagger_rus.zip.
[nltk_data]    | Downloading package basque_grammars to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping grammars/basque_grammars.zip.
[nltk_data]    | Downloading package bcp47 to /root/nltk_data...
[nltk_data]    | Downloading package biocreative_ppi to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/biocreative_ppi.zip.
[nltk_data]    | Downloading package bllip_wsj_no_aux to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping models/bllip_wsj_no_aux.zip.
[nltk_data]    | Downloading package book_grammars to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping grammars/book_grammars.zip.
[nltk_data]    | Downloading package brown to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/brown.zip.
[nltk_data]    | Downloading package brown_tei to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/brown_tei.zip.
[nltk_data]    | Downloading package cess_cat to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/cess_cat.zip.
[nltk_data]    | Downloading package cess_esp to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/cess_esp.zip.
[nltk_data]    | Downloading package chat80 to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/chat80.zip.
[nltk_data]    | Downloading package city_database to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/city_database.zip.
[nltk_data]    | Downloading package cmudict to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/cmudict.zip.
[nltk_data]    | Downloading package comparative_sentences to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/comparative_sentences.zip.
[nltk_data]    | Downloading package comtrans to /root/nltk_data...
[nltk_data]    | Downloading package conll2000 to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/conll2000.zip.
[nltk_data]    | Downloading package conll2002 to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/conll2002.zip.
[nltk_data]    | Downloading package conll2007 to /root/nltk_data...
```

```
import nltk
from nltk.corpus import stopwords


from nltk.tokenize import word_tokenize


data="Data science is the study of data to extract meaningful insights for business. It is a multidisciplinary approach that combines princi


words=word_tokenize(data.lower())
print(words)
```

```
['data', 'science', 'is', 'the', 'study', 'of', 'data', 'to', 'extract', 'meaningful', 'insights', 'for', 'business', '.', 'it', 'is', '
```

```
stop_words=set(stopwords.words('english'))


print(stop_words)
#stopwords list
```

```
 'that', 'where', 'myself', "she's", 'no', 'having', "isn't", 've', 'have', 'there', 'were', 'through', 'down', 'those', 'now', 'wasn',
```

```
filtered_words=[]


for w in words:
  if w not in stop_words:
    filtered_words.append(w)
print(filtered_words)
#The words that aren't stop words are filtered over here
```

```
['data', 'science', 'study', 'data', 'extract', 'meaningful', 'insights', 'business', '.', 'multidisciplinary', 'approach', 'combines',
```

```
stopwords_in_para=[]


for w in words:
  if w in stop_words:
    stopwords_in_para.append(w)
print(stopwords_in_para)
#The words that are stopwords in the para taken are here.
```

```
['is', 'the', 'of', 'to', 'for', 'it', 'is', 'a', 'that', 'and', 'from', 'the', 'of', 'and', 'to', 'of', 'this', 'to', 'and', 'what', 'w
```

```
print(len(stop_words))
```

```
179
```

Start coding or generate with AI.