NAME: ARJUN T ANIL

REG NO: 24PMC117

**CADL1:**

*With a text corpus (e.g., a collection of news articles or social media posts), perform the following pre-processing steps using Python libraries (NLTK, SpaCy):*

*Tokenization, Stemming, Lemmatization, and Stop word removal.*

*Document the code and the results of each pre-processing step and upload their GitHub link in a padlet.*

**GitHub Link :** <https://github.com/arjuntanil/CADL1.git>

**Code:**

**1. Import & Download Resources**

import nltk

nltk.download('punkt')

nltk.download('punkt\_tab')

nltk.download('stopwords')

nltk.download('wordnet')

import spacy

nlp = spacy.load("en\_core\_web\_sm")

* **NLTK** and **spaCy** are two popular NLP libraries.
* punkt → Needed for sentence/word tokenization.
* punkt\_tab → Supports additional tokenization models.
* stopwords → List of common words like *is, the, and, of* etc.
* wordnet → A lexical database required for **lemmatization**.
* spacy.load("en\_core\_web\_sm") → Loads a pre-trained English NLP pipeline (tokenizer, POS tagger, lemmatizer, stopword list, etc.).

**2. Sample Dataset**

corpus = [

"The stock market crashed due to global uncertainty.",

"Natural Language Processing is a key part of Artificial Intelligence.",

"Google releases a new AI model to improve search results.",

"The weather today is sunny and pleasant in New York.",

"Sports events are being postponed because of heavy rains."

]

👉 A **corpus** is just a collection of text documents.  
Here, you created 5 short sentences to test preprocessing.

**3. Print Original Corpus**

print("📌 Original Corpus:")

for i, doc in enumerate(corpus, 1):

print(f"{i}. {doc}")

👉 Displays each sentence with its index before applying NLP steps.

**4. NLTK Preprocessing**

from nltk.tokenize import word\_tokenize

from nltk.corpus import stopwords

from nltk.stem import PorterStemmer, WordNetLemmatizer

stop\_words = set(stopwords.words('english'))

stemmer = PorterStemmer()

lemmatizer = WordNetLemmatizer()

* word\_tokenize → Splits text into words.
* stopwords.words('english') → English stop word list.
* PorterStemmer() → Reduces words to their root form (but not always meaningful).
  + Example: *running → run, studies → studi*
* WordNetLemmatizer() → Uses vocabulary + grammar to reduce to dictionary form.
  + Example: *running → run, studies → study*

**Processing Each Sentence (NLTK)**

for i, doc in enumerate(corpus, 1):

tokens = word\_tokenize(doc.lower())

no\_stop = [w for w in tokens if w.isalpha() and w not in stop\_words]

stemmed = [stemmer.stem(w) for w in no\_stop]

lemmatized = [lemmatizer.lemmatize(w) for w in no\_stop]

1. doc.lower() → Convert sentence to lowercase.
2. word\_tokenize(...) → Breaks into words (tokens).
3. [w for w in tokens if w.isalpha() and w not in stop\_words] → Keeps only alphabetic words & removes stopwords.
   * Example: *"the" → removed, "is" → removed*.
4. stemmer.stem(w) → Applies stemming.
5. lemmatizer.lemmatize(w) → Applies lemmatization.

👉 The print statements then show each stage for every sentence.

**5. spaCy Preprocessing**

for i, doc in enumerate(corpus, 1):

spacy\_doc = nlp(doc.lower())

tokens = [token.text for token in spacy\_doc]

no\_stop = [token.text for token in spacy\_doc if not token.is\_stop and token.is\_alpha]

lemmatized = [token.lemma\_ for token in spacy\_doc if not token.is\_stop and token.is\_alpha]

* nlp(doc.lower()) → Passes sentence to spaCy pipeline.
* token.text → Extracts tokens.
* token.is\_stop → Checks if word is a stopword.
* token.is\_alpha → Ensures only alphabetic tokens (no numbers/punctuations).
* token.lemma\_ → Gets the **lemma** (root dictionary form).

👉 SpaCy doesn’t have a built-in stemmer because **lemmatization is more accurate**.

Output:



