```
!pip install nltk spacy
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.12/dist-packages (3.9.1)
Requirement already satisfied: spacy in /usr/local/lib/python3.12/dist-packages (3.8.11)
Requirement already satisfied: click in /usr/local/lib/python3.12/dist-packages (from nltk) (8.3.1)
Requirement already satisfied: joblib in /usr/local/lib/python3.12/dist-packages (from nltk) (1.5.3)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.12/dist-packages (from nltk) (2025.11.
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-packages (from nltk) (4.67.1)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.12/dist-packages (from spa
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.12/dist-packages (from spa
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.12/dist-packages (from spacy
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.0
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (3
Requirement already satisfied: thinc<8.4.0,>=8.3.4 in /usr/local/lib/python3.12/dist-packages (from spacy) (8.3
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.12/dist-packages (from spacy) (1.
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.5
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.12/dist-packages (from spacy)
Requirement already satisfied: weasel<0.5.0,>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (0.
Requirement already satisfied: typer-slim<1.0.0,>=0.3.0 in /usr/local/lib/python3.12/dist-packages (from spacy)
Requirement already satisfied: numpy>=1.19.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.0.2)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.12/dist-packages (from spacy)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.1.6)
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from spacy) (75.2.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (25.0)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.12/dist-packages (from pydantic
Requirement already satisfied: pydantic-core==2.41.4 in /usr/local/lib/python3.12/dist-packages (from pydantic!
Requirement already satisfied: typing-extensions>=4.14.1 in /usr/local/lib/python3.12/dist-packages (from pydan
Requirement already satisfied: typing-inspection>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from pydant
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from reques
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests<3.0
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests<3.0
Requirement already satisfied: blis<1.4.0,>=1.3.0 in /usr/local/lib/python3.12/dist-packages (from thinc<8.4.0,
Requirement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.12/dist-packages (from thinc<
Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in /usr/local/lib/python3.12/dist-packages (from weas
Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in /usr/local/lib/python3.12/dist-packages (from weasel
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from jinja2->spacy)
Requirement already satisfied: wrapt in /usr/local/lib/python3.12/dist-packages (from smart-open<8.0.0,>=5.2.1-
import nltk
import spacy
```

```
# Medical text (sample, not real patient data)
medical_text = """
Patients with diabetes often require regular monitoring of blood glucose levels.
The treatment includes insulin therapy, dietary control, and physical activity.
Doctors diagnose the condition based on symptoms and laboratory tests.
Early diagnosis and proper management can prevent serious complications.
"""

print(medical_text)
```

```

Patients with diabetes often require regular monitoring of blood glucose levels.
The treatment includes insulin therapy, dietary control, and physical activity.
Doctors diagnose the condition based on symptoms and laboratory tests.
Early diagnosis and proper management can prevent serious complications.
```

```
from nltk.tokenize import sent_tokenize

# Sentence tokenization
sentences = sent_tokenize(medical_text)

print("Sentence Tokens:")
for s in sentences:
    print("-", s)
```

```
Sentence Tokens:
-
Patients with diabetes often require regular monitoring of blood glucose levels.
- The treatment includes insulin therapy, dietary control, and physical activity.
- Doctors diagnose the condition based on symptoms and laboratory tests.
- Early diagnosis and proper management can prevent serious complications.
```

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

```
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.
True
```

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

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

```
from nltk.tokenize import word_tokenize

# Word tokenization using NLTK
words_nltk = word_tokenize(medical_text)

print("NLTK Word Tokens:")
print(words_nltk)
```

```
iabetes', 'often', 'require', 'regular', 'monitoring', 'of', 'blood', 'glucose', 'levels', '.', 'The', 'treatme
```

```
# Load spaCy model
nlp = spacy.load("en_core_web_sm")

# Tokenization using spaCy
doc = nlp(medical_text)
words_spacy = [token.text for token in doc]

print("spaCy Word Tokens:")
print(words_spacy)
```

```
spaCy Word Tokens:
['\n', 'Patients', 'with', 'diabetes', 'often', 'require', 'regular', 'monitoring', 'of', 'blood', 'glucose', '
```

```
from nltk.stem import PorterStemmer

# Initialize Porter Stemmer
stemmer = PorterStemmer()

# Apply stemming
stemmed_words = [stemmer.stem(word) for word in words_nltk if word.isalpha()]

print("Stemmed Words:")
print(stemmed_words)
```

```
Stemmed Words:
['patient', 'with', 'diabet', 'often', 'requir', 'regular', 'monitor', 'of', 'blood', 'glucos', 'level', 'the',
```

```
# Initialize WordNet Lemmatizer
lemmatizer = WordNetLemmatizer()

lemmatized_words_nltk = [lemmatizer.lemmatize(word) for word in words_nltk if word.isalpha()]

print("NLTK Lemmatized Words:")
print(lemmatized_words_nltk)
```

```
NLTK Lemmatized Words:
['Patients', 'with', 'diabetes', 'often', 'require', 'regular', 'monitoring', 'of', 'blood', 'glucose', 'level'
```

```
lemmatized_words_spacy = [token.lemma_ for token in doc if token.is_alpha]

print("spaCy Lemmatized Words:")
print(lemmatized_words_spacy)
```

```
spaCy Lemmatized Words:
['patient', 'with', 'diabete', 'often', 'require', 'regular', 'monitoring', 'of', 'blood', 'glucose', 'level',
print("Word\t\tStemmed\t\tLemmatized")
print("-"*50)

for word in ["diagnosis", "patients", "treated", "monitoring", "includes"]:
    print(f"{word}\t\t{stemmer.stem(word)}\t\t{lemmatizer.lemmatize(word)}")
```

```
Word            Stemmed         Lemmatized
--------------------------------------------------
diagnosis               diagnosi                diagnosis
patients                patient         patient
treated         treat           treated
monitoring              monitor         monitoring
includes                includ          includes
```