#part1

sample\_doc = "Natural Language Processing (NLP) is a sub-field of artificial intelligence concerned with understanding and processing human language."

import nltk

import string

from nltk.corpus import stopwords

from nltk.tokenize import word\_tokenize

from nltk import pos\_tag

from nltk.stem import PorterStemmer, WordNetLemmatizer

from sklearn.feature\_extraction.text import TfidfVectorizer

import pandas as pd

# Download required resources

nltk.download('punkt')

nltk.download('stopwords')

nltk.download('averaged\_perceptron\_tagger')

nltk.download('wordnet')

nltk.download('omw-1.4')

nltk.download('punkt\_tab')

nltk.download('averaged\_perceptron\_tagger\_eng')

#tokenise

tokens = word\_tokenize(sample\_doc)

print("Tokens:", tokens)

#pos tagging

pos\_tags = pos\_tag(tokens)

print("POS Tags:", pos\_tags)

#stop-words removal

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

filtered\_tokens = [word for word in tokens if word.lower() not in stop\_words and word.isalpha()]

print("Filtered Tokens:", filtered\_tokens)

#stemming

stemmer = PorterStemmer()

stemmed\_tokens = [stemmer.stem(word) for word in filtered\_tokens]

print("Stemmed Tokens:", stemmed\_tokens)

#lemmatizing

lemmatizer = WordNetLemmatizer()

lemmatized\_tokens = [lemmatizer.lemmatize(word) for word in filtered\_tokens]

print("Lemmatized Tokens:", lemmatized\_tokens)

#part2

#TF-IDF Representation

documents = [

"Natural Language Processing helps computers understand human language.",

"Artificial Intelligence includes machine learning and NLP.",

"Text preprocessing includes tokenization, stemming, and lemmatization."

]

# Initialize TF-IDF Vectorizer

vectorizer = TfidfVectorizer()

# Fit and transform documents

tfidf\_matrix = vectorizer.fit\_transform(documents)

# Convert to DataFrame for readability

tfidf\_df = pd.DataFrame(tfidf\_matrix.toarray(), columns=vectorizer.get\_feature\_names\_out())

print(tfidf\_df)