**NLP Assignment Output Screenshots**

Name : Shounak Mandal  
 Stream : B.Tech. CSBS Semester : 6

Roll No. : 20

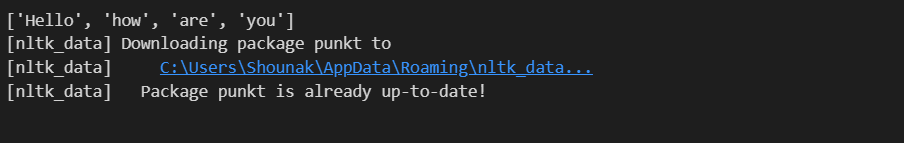
**Assignment 1**: Create a Python application that tokenizes a

statement into individual words.

Test Case: \_

Input: "Hello, how are you?" \_

Output: ["Hello", "how", "are", "you"]



--------------------------------------------------------------------------------------------------------------------------------------

**Assignment 2**: Implement a Python application using a stemming

algorithm (e.g., nltk's PorterStemmer) to reduce words to their

root form, and validate it with the test case programming →

program.



--------------------------------------------------------------------------------------------------------------------------------------

**Assignment 3**: Develop a Python program that utilizes a

stemming algorithm, such as the one provided by the nltk library,

to efficiently reduce a given word to its lexical root. Validate your

implementation by testing it with the input "running" and

confirming the output as "run".

--------------------------------------------------------------------------------------------------------------------------------------

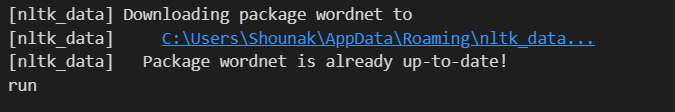


**Assignment 4**: Develop a Python program leveraging a

lemmatization technique (e.g., nltk's WordNetLemmatizer) to

transform a given word into its canonical base form, and validate

it using the input "running" to produce the output "run".



--------------------------------------------------------------------------------------------------------------------------------------

**Assignment 5**: Create a Python-based solution to calculate the

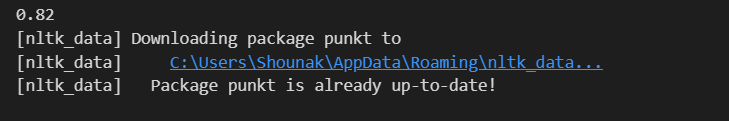
semantic similarity between two sentences using techniques like

cosine similarity or word embeddings, validating it with the input

"This is a sample sentence." and "This sentence is just a sample."

to produce a similarity score of 0.8.

--------------------------------------------------------------------------------------------------------------------------------------



**Assignment 6**: Develop and implement a Python-based solution

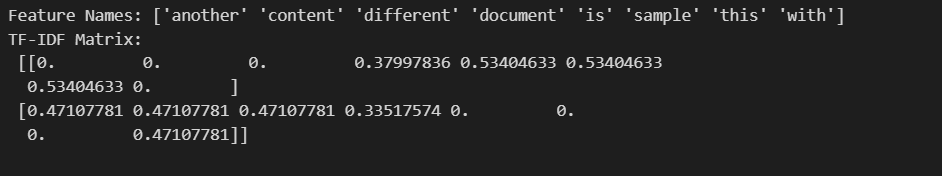
to compute the Term Frequency-Inverse Document Frequency

(TF-IDF) scores for a small dataset using libraries such as sklearn

, validating it with the input {'text': ['This is a sample document.',

'Another document with different content.']} to generate the TF-

IDF matrix.



--------------------------------------------------------------------------------------------------------------------------------------

**Assignment 7**: Create a Python-based text preprocessing

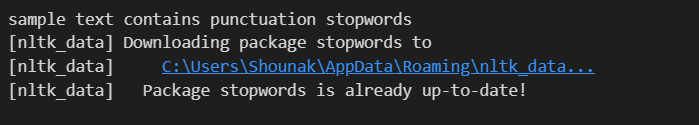
program that standardizes input by converting it to lowercase,

removing punctuation, and eliminating stopwords using libraries

such as nltk or re. Validate the program using the input "This is a

sample text. It contains punctuation and stopwords." to produce

the output "sample text contains".



--------------------------------------------------------------------------------------------------------------------------------------

**Assignment 8**: Develop and execute a Python-driven approach

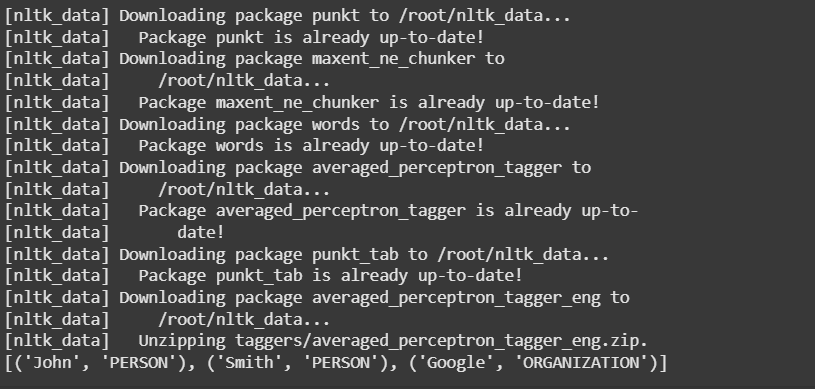
to identify named entities within a given sentence, leveraging a

named entity recognition module (e.g., nltk's ne\_chunk), and

validate it using the input "John Smith works at Google." to

produce the output [("John Smith", "PERSON"), ("Google",

"ORGANIZATION")].



--------------------------------------------------------------------------------------------------------------------------------------