Lab 10 :

**Download NLTK**

Pip install nltk

**Download all these libraries in PYTHON IDLE SHELL**

Execute each line separately

import nltk

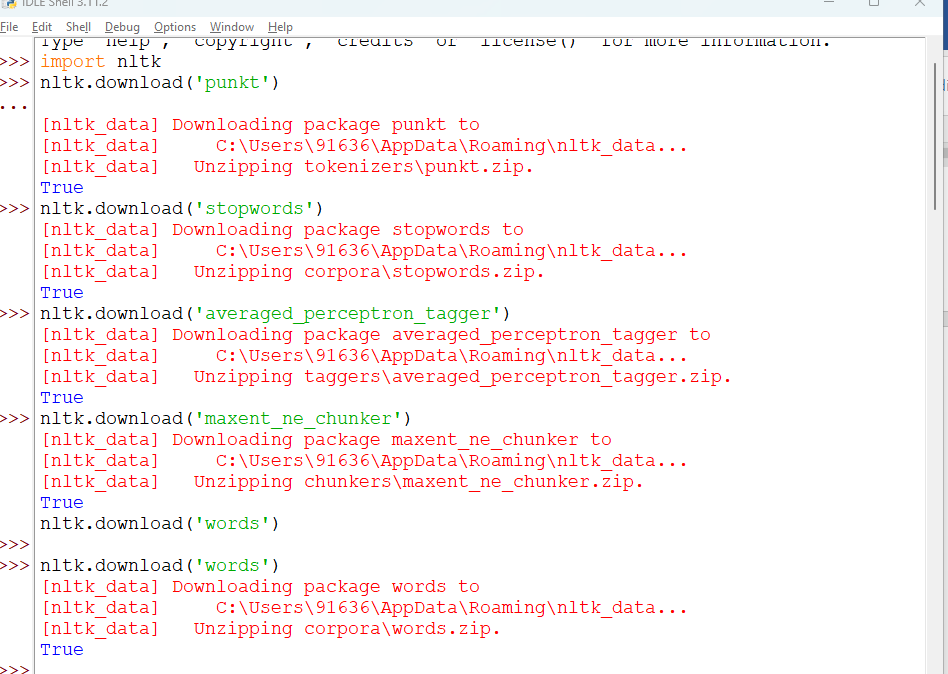
nltk.download('punkt')

nltk.download('stopwords')

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

nltk.download('maxent\_ne\_chunker')

nltk.download('words')



program:

import nltk

from nltk.tokenize import word\_tokenize

from nltk.corpus import stopwords

from nltk.stem import PorterStemmer

from nltk import pos\_tag, RegexpParser, ne\_chunk

# Sample Text

text = "change is a contant thing "

# Tokenizing

tokens = word\_tokenize(text)

# Filtering Stop Words

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

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

# Stemming

stemmer = PorterStemmer()

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

# Part of Speech Tagging

pos\_tags = pos\_tag(tokens)

# Chunking

grammar = r"""

NP: {<DT>?<JJ>\*<NN>} # Chunk Noun Phrases

VP: {<VB.\*><DT>?<JJ>\*<NN|RB|VB.\*>\*<RB>?} # Chunk Verb Phrases

"""

chunk\_parser = RegexpParser(grammar)

chunked = chunk\_parser.parse(pos\_tags)

# Named Entity Recognition (NER)

ner\_tags = ne\_chunk(pos\_tags)

# Displaying Results

print("Original Text:", text)

print("\nTokenization:")

print(tokens)

print("\nFiltered Stop Words:")

print(filtered\_tokens)

print("\nStemmed Tokens:")

print(stemmed\_tokens)

print("\nPart of Speech Tagging:")

print(pos\_tags)

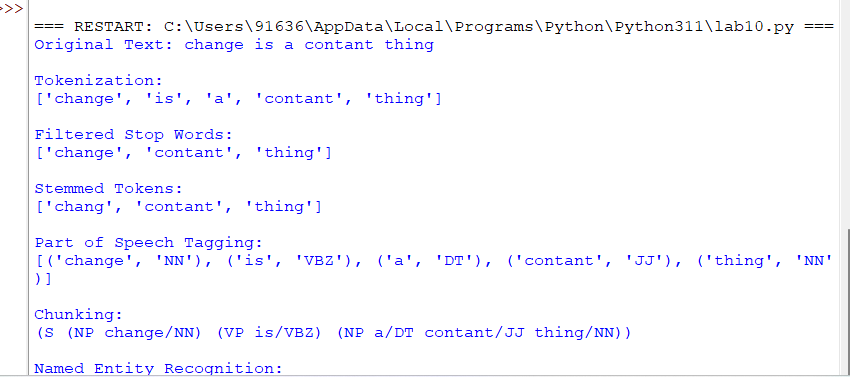
print("\nChunking:")

print(chunked)

print("\nNamed Entity Recognition:")

print(ner\_tags)

**output:-**

****