```
# OUTPUT 1-
```

```
['Hello', 'Keshav', ', 'Good', 'Morning', '!']
```

05 07 24	Poornima University
#	EXPERIMENT - 01
*	AIM :- Write a program to tokenize the sentence into words for the further analysis.
*	PROGRAM 1-
	import nitk nitk.download C'punkt') from nitk.tokenize import word-tokenize
	text = "Hello Keshav, Good Morning!" word-tokenize (text)
*	EXPLAINATION 1-
0	Tokenization is the process of converting a sequence of text into smaller parts, known as tokens. In this program, tokens are the individual words in the
0	The word-tokenize function uses 'punkt' models to find where words start and end, handling punctuation enrectly.
	Page No. 1

OUTPUT :-

Original: -> Hello Keshav, the NLP model is 98% accurate.

Normalized: -> Hello Keshav the nlp model is minety - eight accurate

EXPERIMENT - 02

AIM 1- Write a program to normalize the sentence to eliminate the unwanted punctuation, converting into lower case or upper case of the entire document, expanding abbreviation, numbers into words and canonicalization.

* PROGRAM !-

from num2 words import num2 words

def normalize - sentence (sentence):

#Remove punctuation

sentence = re-sub (r'[1/w/s]', " sentence)

convert to lowercase

Sentence = Sentence. lowereage ()

convert numbers to words

words = []

for word in sentence split():

if word is digit ():

words, append (num 2 words (int (word)))

else:

words, append (word)

Join words back into a sentence Sentence = ' join (words

return Sentence

Page No. &

Poornima	University
----------	------------

text = "Hello Keshav, the NLP model is 98% accurate." normalized - text = normalize-sentence (text) print ("Original: -> ", text) print ("Normalized :-> " normalized-text)

Page No. 3

OUTPUT

Method 1: List as an Iterable
Hello
there
SAM

method 2: Using Range Hello there

Method 3: Using Enumerator

Index 0: Hello

Index 1: there

Index 2: SAM

18 07 24	Poornima University	
#	EXPERIMENT - 03	71.90 #
*	AIM :- write a program that splits the following string "Hello there sam" into list and iterate of	Ver
0	List as a Iterable	
р	Using Range An enumerator	
*	PROGRAM 1-	
	# Splitting the string into a list string = "Hello there SAM" string-list = String. split() # return list	
	# 1: List as an Iterable print ("Method 1: List as an Iterable") for item in string-list:	
	print Citem) #2>	
	print ("In Method 2: Using Range") for i in range (len (String-list)): print (String-list [i])	
	print (In Method 3: Using Enumerator") for index, item in enumerate (string-list print (f" Index & index 3: & item?")):
	F	Page No. 4

26/07/24	Poornima University
#	EXPERIMENT - 04
*	AIM: - Convert the following sentence into tokens "NLP is Fun, you must learn it" into lowercase Without splitting With splitting
*	PROGRAM !-
	# Without Splitting S = "NLP is Fun, you must learn it" print (5.lower ())
	# with splitting print Cs. lower (). split())
	print C 5. 10wer C J. Spile C J
	Page No.5.

CUTPUT

mlp is fun, you must learn it

['mlp', 'is', 'fun', ', 'you', 'must', 'learn', 'it']

19/08/24	Poornima University
#	EXPERIMENT - 05
*	AIM! - Write a program to Get the word cloud for the yelp Review data set.
*	Program: import pandas as pd from word cloud import Word Cloud import matphothib.pyplot as plt from nltk.corpus import stopwords import string import nltk # Load the dataset df = pd.read-csu C'amazon-alexa.tsu', sep = "It") print (df)
	# download stopwords mltk.download C'stopwords') # Pre process the text def pre process_text C text): # Convert text to lowercase text = str(text) text = text.lower() # Remove Punctuation text = text.translate Cstr.maketrans(", ", string. punctuation))
	Page No6

Poornima University

```
# Remove stopwords

stop-words = set (stopwords words ('english'))

text = ''.join (word for word in text.split())

word not in stop-words)

return text
```

Text Preprocessing

df ['cleaned-text'] = df ['verified-reviews'].apply

Cpre process-text)

text-combined = ''-join (df ['cleaned-text'].tolist())

Word cloud generation & display

word cloud = Word Cloud (width = 800, height = 400,

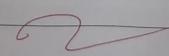
background - color = 'white').generate (text-combined)

plt. figure (figsize = (zo,5))

plt. imshow (word cloud, interpolation = bilinear)

plt. axis ('off')

plt. show()



23/08/24	Poornima University
#	EXPERIMENT - 06
*	AIM i- Write a program for Amazom review dataset to find the maximum number of words used. Get the output for the frequently occurred word in the given data? And also visualize the test data
*	Program: import pandas as pd import re from collections import counter import matplotlib.pyplot as plt from worddoud import WordCloud
	# Load the dataset df = pd. read - csv ('amazon-alexa. tsv', delimiter = 'lt')
	# Preprocess the text data def preprocess-text (text): text = str (text) text = re.sub (r'lw', '', text) #Remove special charaters text = re.sub (r'ls+', '', text) #Replace multi. spaces text = text.lower (). strip() return text
	df ['cleaned - reviews'] = df ['verified - reviews']. apply (preprocess - text) Page No

Poornima University

```
# Find the most frequent word & its count
 all-words = ". join (df ['cleaned_reviews']). split()
 word-counts = Counter (all-words)
 most-common-word, most-common-count = word-counts.
 most-common (1) [0]
print (f" The most frequent word is & most - common-
 word ?' with & most-common-count? occurrences ")
# Visualize the top 10 most frequent words
top-10-words = word-counts. most-common (10)
words, counts = zip (* top-10-words)
plt. figure (figsize = (ZO, 5))
plt. bar (Words, counts, color = 'skyblue')
pit. title ('Top 10 Most Frequent Words in
Amazon Reviews ')
plt, xlabel ('Words')
pit-ylabel ('Frequency')
plt. show()
# Visualize the word cloud
wordcloud = word Cloud Cwidth = 800, height = 400,
background-color = 'white') generate (''.join Call-
words ))
```

Poornima University
plt. figure (figsize = (10,5)) plt. imshow (word cloud, interpolation = bilinear)
plt. show()