OUTPUT :-

['Hello', 'Keshav', ',', 'Good', 'morning', '!']

EXPERIMENT - 01

* AIM :- Write a program to tokenize the sentence into words for the further analysis.

* PROGRAM :-

```
import nltk
nltk.download ('punkt')
from nltk.tokenize import word_tokenize

text = "Hello Keshav, Good morning!"
word_tokenize (text)
```

* EXPLAINATION :-

- 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 sentence.
- The word_tokenize function uses 'punkt' models to find where words start and end, handling punctuation correctly.

OUTPUT :-

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

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

2/07/24

EXPERIMENT - 02

* AIM :- 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 :-

```
import re
from num2words import num2words

def normalize_sentence(sentence):
    #Remove punctuation
    sentence = re.sub(r'[^\w\s]', '', sentence)
    #convert to lowercase
    sentence = sentence.lowercase()
    #convert numbers to words
    words = []
    for word in sentence.split():
        if word.isdigit():
            words.append(num2words(int(word)))
        else:
            words.append(word)
    # Join words back into a sentence
    sentence = ' '.join(words)
    return sentence
```

```
text = "Hello Keshav, the NLP model is 98% accurate."
normalized_text = normalize_sentence(text)

print("Original :-> ", text)
print("Normalized :-> ", normalized_text)
```

OUTPUT

Method 1: List as an Iterable
Hello
there
SAM

Method 2: Using Range
Hello
there
SAM

Method 3: Using Enumerator
Index 0 : Hello
Index 1 : there
Index 2 : SAM

EXPERIMENT - 03

* AIM :- write a program that splits the following string "Hello there SAM" into list and iterate over the list using 3 different methods
  - List as a Iterable
  - Using Range
  - An enumerator

* PROGRAM :-

```
# 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(item)
#2 >
print("In Method 2: Using Range")
for i in range(len(string_list)):
    print(string_list[i])
#3 >
print(In Method 3: Using Enumerator")
for index, item in enumerate(string_list):
    print(f"Index {index}: {item}")
```

26/07/24

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 ( S.lower())

# With Splitting
print (S.lower().split())
```

OUTPUT

nlp is fun, you must learn it

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

EXPERIMENT - 05

* **AIM :-** Write a program to Get the word cloud for the yelp Review data set.

* **PROGRAM :-**

```
import pandas as pd
from wordcloud import WordCloud
import matplotlib.pyplot as plt
from nltk.corpus import stopwords
import string
import nltk

# Load the dataset
df = pd.read_csv('amazon_alexa.tsv', sep="\t")
print(df)

# download stopwords
nltk.download('stopwords')

# Preprocess the text
def preprocess_text(text):
    # Convert text to lowercase
    text = str(text)
    text = text.lower()
    # Remove Punctuation
    text = text.translate(str.maketrans('', '', string.
    punctuation))
```

```
# Remove stopwords
stop_words = set(stopwords.words('english'))
text = ' '.join(word for word in text.split() if
word not in stop_words)
return text

# Text Preprocessing
df['cleaned_text'] = df['verified_reviews'].apply
(preprocess_text)
text_combined = ''.join(df['cleaned_text'].tolist())

# Word cloud generation & display
wordcloud = WordCloud(width = 800, height = 400,
background_color = 'white').generate(text_combined)

plt.figure(figsize = (10, 5))
plt.imshow(wordcloud, interpolation = 'bilinear')
plt.axis('off')
plt.show()
```

EXPERIMENT - 06

* AIM :- Write a program For Amazon 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='\t')

# Preprocess the text data
def preprocess_text(text):
    text = str(text)
    text = re.sub(r'\W', ' ', text) #Remove Special charaters
    text = re.sub(r'\s+', ' ', text) #Replace multi. spaces
    text = text.lower().strip()
    return text

df['cleaned_reviews'] = df['verified_reviews'].
apply(preprocess_text)
```

```
# 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=(10,5))
plt.bar(words, counts, color='skyblue')
plt.title('Top 10 Most Frequent Words in
Amazon Reviews')
plt.xlabel('Words')
plt.ylabel('Frequency')
plt.show()

#Visualize the word cloud
wordcloud = wordCloud(width=800, height=400,
background_color='white').generate(''.join(all_
words))
```

```
plt.figure (figsize = (10,5))
plt.imshow (word cloud , interpolation = 'bilinear')
plt.axis ('off')
plt.show()
```

2