IARE INSTITUTE OF AERONAUTICAL ENGINEERING

Dundigal, Hyderabad - 500 043
Ph: 8886234501, 8886234502
iare.ac.in

LABORATORY WORK SHEET

Date: 03/02/2025

Roll No: 22951A6661 Name: B. Vaishnavi

Exp No: 02 Experiment Name: Text Preprocessing in Python

DAY TO DAY EVALUATION:

	Preparation	Algorithm	Source Code	Program Execution	Viva voce	Total
		Performance in the Laboratory	Calculations and Graphs	Results and Error Analysis		
Max. Marks	4	4	4	4	4	20
Obtained	4	4	4	4	4	20

Signature of Lab I/C

START WRITING FROM HERE:

2.1 Building NLP model and perform text processing
prepare the text data for the NLP model building and perform the text pre-processing. Use the required pre-processing steps based on the dataset prepared and understand the steps involved in text pre-Processing.

Source Code:-

```
import numpy as np
import pandas as pd
import re
~~for ai~~
import string
import inflect
def preprocess_text(text):
    text = text.lower()
    p = inflect.engine()
    def Convert_number_to_words(match):
        return p.number_to_words(match.group())
    text = re.sub(r'\d+', convert_number_to_words, text)
    text = text.translate(str.maketrans('', '', string.punctuation)
    text = ' '.join (text.split())
    return text
statements = [
     "Hello ! How are you Today ? ",
     "I have 2 cats and 1. dog.",
     "NLP is amazing!!!",
     "The temperature is 30 degrees celsius."
  ]
preprocessed _statements = [preprocess_text (Sentence) for Sentence
                in Statements]
for original , processed in zip (statements , preprocessed_ statements).
    Print (f"Original : {original } \n Preprocessed : {processed}\n")
```

Output:-

```
Original: Hello! How are you today?
Preprocessed: hello how are you today
Original: I have 2 cats and 1 dog.
Preprocessed: i have two cats and one dog
Original: NLP is amazing!!!
Preprocessed NLP is amazing
original: The temperature is 30 degrees celsius
Preprocessed: the temperature is 30 degree celsius
```

2.2 TEXT PREPROCESSING OPERATIONS:

Prepare the text data for the NLP model building and perform the text pre-processing. Use the required pre-processing steps.

Source Code:-

```
import pandas as pd
import numpy as np
import spacy
from spacy.lang. en.stop words import STOP_WORDS as stop.words.
df = pd.read_csv ('https://raw.githubuserContent.Com/laxmimerit/csv', encoding=
                                                                  'latin1')
df ['word_counts'] = df ['twitts'].apply(lambda x: len (str(x).split()))
df ['char_counts'] = df ['twitts'].apply (lambda x : len ( str(x)))
df ['avg_word_length'] = df ['char_counts'] / df ['word_counts']
print ("Sample Data:")
print (df.sample (5))
Print ("\nMax Word Count : ", df ['word_counts '].max ())
Print ("Min Word Count :", df['word-Counts'].min())
Print ("\nTweets with one word:")
print (df[df['word-counts'] = = 1])
```

output:-

```
Max word Count :32
Min word Count : 1
Tweets with One word :
                -twits   Sentiment   word-Counts  Char_Counts )
385                                      1           9
691          homework        0           1           9
1104        @emrelly         0           1           13
1286      disappointed       0           1           16
1325   @officialmgnfox       0           1           9
             headache        0
....
2947     8.0
3176     13.0
```

2.3 Preprocessing and cleaning.

Prepare the text data for NLP model building and perform the pre-processing.

Source code:-

```
import pandas as pd
import numpy as np
import spacy
import re
from spacy.lang.en.stop words import STOP_WORDS as stopwords
from contractions import fix
df = pd.read-csv("https://raw.githubuserContent.com/laxmierit/twitter -
      data/master/twitter4000.csv', encoding ='latin1')
def preprocess-Text (text):
    if not isinstance (text, str):
        return ""
    Text = text.lower()
    text = fix(text)
    text = re.sub(r'\s+@\s+', '', text)
    return Text
df['clean-Text']= df['twits'].astype (str).apply (preprocess_text)
print ("\nEmail count :", df['clean-text'])
df['email-count]= df['twits'].apply(lambda x:len(re.findall (r'\s+@\s+',
                                                       str(x))))
Print ("\nEmail Count:", df ['email_Count"])
```

Output:

```
Email Count : 0
1     0
2     0
3     0
4     0
      ..
3995  0
3996  0
3997  0
3998  0
3999  0
...
2292  in line for the simpsons ride.more cute foreign.. 0
1311  need to save up for this sexail $70 dress ..g...    0
```

2.4 Preprocessing and cleaning.

implement the text pre processing and perform whiroes operations

Source Code :-

```
import pandas as pd
import re unicodedata nltk
from bs4 import Beautifulsoup
from nltk.Corpus import stopwords
nltk.download ('stopwords')
print (df.columns)
def remove_html-tags (twitts):
    return Beautifulsoup (twitts, 'html.parser').get-text()
def remove_accented-chars(text):
    return ''.join (c for c in unicodedata.normalize ('NFKD', text) if not
                                unicodedata.Combining(c))
def remove-stopwords (text):
    stop-words = set (stopwords.words ('english'))
    return ''.join (word for word in words if word.lower() not in stop-words)
df['cleaned-text'] = df ['twitts'].astype (str)
df['cleaned-text'] = df ['cleaned-text'].apply (remove_html_tags)
df['cleaned-text'] = df ['cleaned_text'].apply (remove-accented_chars)
df['cleaned-text'] = df['cleaned_text'].apply (remove-stopwords)
df['cleaned-text', 'twitter 4000, cleaned.csv', index = false
df.to-csv ('twitter 4000, cleaned.csv', index = false)
print ("Text preprocessing completed. cleaned data saved to twitter 4000_
                                                        cleaned.csv!")
```

output:-

```
Text preprocessing Completed. Cleaned data. Saved to
        twitter4000-cleaned.csv'.
[nltk_data] Downloading package stopwords to /usr/share /nltk_data..
[nltk-data] package stopwords is already up-to-date!
```

2.5 Preprocessing and cleaning:

Prepare the text data for the NLPmodel building and perform text preprocessing

Source Code:

```
import pandas as pd
import re unicodedata nltk
from bs4 import BeautifulSoup
from nltk.corpus import stopwords
from textblob import TextBlob
from wordcloud import WordCloud
nltk.download ('stopwords')
df =pd.read-csv ('https://raw.githubusercontent.com/laxmimit/twitter.csv',encoding
                                                                 ='latin1')

def Correct-spelling (text):
     return str(TextBlob (text) . Correct())
def tokenize-text (text):
     return TextBlob (text).words
df ['cleaned-text'] = df['cleaned-text'].apply (corrected_Spelling)
df ['tokens'] = df['cleaned-text'].apply (tokenize -Text)
def generate-word_cloud (text):
     wordcloud = WordCloud (width =800, height = 400 , bg-clr = 'white').generate
                                                          (' '.join (text))
     plt.figure (figsize=(10,5))
     plt. imshow (word cloud , interpolation ='bilinear')
     plt. axis ('off')
     plt.show()
generate-word-cloud(df['cleaned-text'])
df.to_csv ('twitter4000-Cleaned.csv', index = False)
print ("Text Preprocessing Completed. cleaned data saved to 'twitter4000-
                                         cleaned.csv'.")
```

Output:

```
[nltk_data] Downloading package stopwords to /usr/share/nltk_data..
[nltk_data] package stopword is already up-to-date!
```