```
import pandas as pd
df = pd.read_excel(r'C:\Users\LENOVO\Desktop\NLP\Internshala NLP
Assignment 2\text_docs.xlsx')
print(df)
```

```
   document_id                                               text
0            1  The stock market has been experiencing volatil...
1            2  The economy is growing, and businesses are opt...
2            3  Climate change is a critical issue that needs ...
3            4  Advances in artificial intelligence have revol...
4            5  The rise of electric vehicles is shaping the f...
5            6  Healthcare is evolving with the introduction o...
6            7  The entertainment industry is shifting towards...
7            8  Social media is influencing the way people int...
8            9  Governments around the world are investing in ...
9           10  Cybersecurity is an ongoing concern as digital...
```

Task 1

```
# Total no. of rows and columns
df.shape
```

```
(10, 2)
```

```
# no of unqiue documents
df.nunique()
```

```
document_id    10
text           10
dtype: int64
```

Preprocessing steps

```
# Now I will do the pre-process steps
df.drop_duplicates(inplace=True)
```

```
df.isnull().sum()
```

```
document_id    0
text           0
dtype: int64
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 2 columns):
 #   Column       Non-Null Count  Dtype
```

```
---  ------       --------------  -----
 0   document_id  10 non-null     int64
 1   text         10 non-null     object
dtypes: int64(1), object(1)
memory usage: 292.0+ bytes

 # removing spaces from columns names

df.columns = df.columns.str.strip()

 # removing spaces from columns
df['text'] = df['text'].str.strip()

# 1. Convert all text to lowercase
# creating a function to transform into lower case

def low(df):
   for i in df.columns:
       if df[i].dtypes == 'object':
           df[i] = df[i].str.lower()
   return df

low(df)

   document_id                                               text
0            1  the stock market has been experiencing volatil...
1            2  the economy is growing, and businesses are opt...
2            3  climate change is a critical issue that needs ...
3            4  advances in artificial intelligence have revol...
4            5  the rise of electric vehicles is shaping the f...
5            6  healthcare is evolving with the introduction o...
6            7  the entertainment industry is shifting towards...
7            8  social media is influencing the way people int...
8            9  governments around the world are investing in ...
9           10  cybersecurity is an ongoing concern as digital...

# 2. Remove punctuation and special characters.
import re #(regex)
df['text'] =  df['text'].apply(lambda x: re.sub(r'[^A-Ba-z\s]', '',
x)) # ^A-za-z, it means remove everthing except A-Z and a-z
# \ is an excape character which tells python to treat the character
beside it as a special character. So \s means, single space.
#So im telling python remove anything except alphabets from A to Z and
single space
print(df)

   document_id                                               text
0            1  the stock market has been experiencing volatil...
1            2  the economy is growing and businesses are opti...
2            3  climate change is a critical issue that needs ...
3            4  advances in artificial intelligence have revol...
```

```
4            5  the rise of electric vehicles is shaping the f...
5            6  healthcare is evolving with the introduction o...
6            7  the entertainment industry is shifting towards...
7            8  social media is influencing the way people int...
8            9  governments around the world are investing in ...
9           10  cybersecurity is an ongoing concern as digital...
```

```python
# tokenizing
from nltk.tokenize import word_tokenize
df['tokens'] = df['text'].apply(lambda x: word_tokenize(x.lower()))
print(df)
```

```
   document_id                                               text  \
0            1  the stock market has been experiencing volatil...
1            2  the economy is growing and businesses are opti...
2            3  climate change is a critical issue that needs ...
3            4  advances in artificial intelligence have revol...
4            5  the rise of electric vehicles is shaping the f...
5            6  healthcare is evolving with the introduction o...
6            7  the entertainment industry is shifting towards...
7            8  social media is influencing the way people int...
8            9  governments around the world are investing in ...
9           10  cybersecurity is an ongoing concern as digital...

                                              tokens
0  [the, stock, market, has, been, experiencing, ...
1  [the, economy, is, growing, and, businesses, a...
2  [climate, change, is, a, critical, issue, that...
3  [advances, in, artificial, intelligence, have,...
4  [the, rise, of, electric, vehicles, is, shapin...
5  [healthcare, is, evolving, with, the, introduc...
6  [the, entertainment, industry, is, shifting, t...
7  [social, media, is, influencing, the, way, peo...
8  [governments, around, the, world, are, investi...
9  [cybersecurity, is, an, ongoing, concern, as, ...
```

```python
from nltk.corpus import stopwords
stop_words = set(stopwords.words('English'))
df['after_stop'] = df['tokens'].apply(lambda x: [i for i in x if i not 
in stop_words])
print('Before stopwatch: ', df['tokens'].head())
print('After stopwatch: ', df['after_stop'].head())
```

```
Before stopwatch:  0    [the, stock, market, has, been, 
experiencing, ...
1    [the, economy, is, growing, and, businesses, a...
2    [climate, change, is, a, critical, issue, that...
3    [advances, in, artificial, intelligence, have,...
4    [the, rise, of, electric, vehicles, is, shapin...
Name: tokens, dtype: object
```

```
After stopwatch:  0    [stock, market, experiencing, volatility,
due,...
1    [economy, growing, businesses, optimistic, fut...
2    [climate, change, critical, issue, needs, imme...
3    [advances, artificial, intelligence, revolutio...
4    [rise, electric, vehicles, shaping, future, au...
Name: after_stop, dtype: object
```

```python
# applying lemmatization with POS tags
from nltk import pos_tag
from nltk.corpus import wordnet
from nltk.stem import WordNetLemmatizer

df['pos_tags'] = df['after_stop'].apply(lambda x: pos_tag(x))

def get_wordnet_pos(tag):
    if tag.startswith('J'): return wordnet.ADJ
    elif tag.startswith('V'): return wordnet.VERB
    elif tag.startswith('N'): return wordnet.NOUN
    elif tag.startswith('R'): return wordnet.ADV
    else: return wordnet.NOUN

lemm = WordNetLemmatizer()
df['lemm_with_pos'] = df['pos_tags'].apply(lambda x:
[lemm.lemmatize(word, get_wordnet_pos(pos)) for (word, pos) in x])
print(df[['after_stop', 'lemm_with_pos']].head())
```

```
                                          after_stop  \
0  [stock, market, experiencing, volatility, due,...
1  [economy, growing, businesses, optimistic, fut...
2  [climate, change, critical, issue, needs, imme...
3  [advances, artificial, intelligence, revolutio...
4  [rise, electric, vehicles, shaping, future, au...

                                       lemm_with_pos
0  [stock, market, experience, volatility, due, e...
1      [economy, grow, business, optimistic, future]
2  [climate, change, critical, issue, need, immed...
3  [advance, artificial, intelligence, revolution...
4  [rise, electric, vehicle, shape, future, autom...
```

```python
df['pos_tags'] # as we can see Pos tags Identifies the grammatical
role of each word (noun, verb, adjective, etc.)
```

```
0    [(stock, NN), (market, NN), (experiencing, VBG...
1    [(economy, NN), (growing, VBG), (businesses, N...
2    [(climate, NN), (change, NN), (critical, JJ), ...
3    [(advances, NNS), (artificial, JJ), (intellige...
4    [(rise, NN), (electric, JJ), (vehicles, NNS), ...
5    [(healthcare, NN), (evolving, VBG), (introduct...
6    [(entertainment, NN), (industry, NN), (shiftin...
7    [(social, JJ), (media, NNS), (influencing, VBG...
8    [(governments, NNS), (around, IN), (world, NN)...
9    [(cybersecurity, NN), (ongoing, VBG), (concern...
Name: pos_tags, dtype: object
```

```
for i in df['after_stop']:
       x = pos_tag(i)
       for raj, ratnajit in x:
          r=  get_wordnet_pos(ratnajit)
          print(f'Done Raj {lemm.lemmatize(raj, pos=r)}')

# WordNetLemmatizer expects POS tags in WordNet format:
# N (noun), V (verb), A (adjective), R (adverb).
# But NLTK's pos_tag() returns Penn Treebank tags like NN, VBG, JJ,
RB.
# So we must convert these Penn tags to WordNet tags before
lemmatizing.
# Without correct POS, words like 'experiencing' will NOT lemmatize to
'experience'.
```

```
Done Raj stock
Done Raj market
Done Raj experience
Done Raj volatility
Done Raj due
Done Raj economic
Done Raj uncertainty
Done Raj economy
Done Raj grow
Done Raj business
Done Raj optimistic
Done Raj future
Done Raj climate
Done Raj change
Done Raj critical
Done Raj issue
Done Raj need
Done Raj immediate
Done Raj global
Done Raj attention
Done Raj advance
Done Raj artificial
Done Raj intelligence
Done Raj revolutionize
Done Raj industry
Done Raj worldwide
Done Raj rise
Done Raj electric
Done Raj vehicle
Done Raj shape
Done Raj future
Done Raj automobile
Done Raj industry
Done Raj healthcare
Done Raj evolve
Done Raj introduction
Done Raj new
Done Raj technology
Done Raj treatment
Done Raj entertainment
Done Raj industry
Done Raj shift
Done Raj towards
Done Raj digital
Done Raj streaming
Done Raj platform
Done Raj social
Done Raj medium
Done Raj influence
Done Raj way
Done Raj people
Done Raj interact
Done Raj government
Done Raj around
Done Raj world
Done Raj invest
Done Raj renewable
Done Raj energy
Done Raj project
Done Raj cybersecurity
Done Raj ongoing
Done Raj concern
Done Raj digital
Done Raj platforms
Done Raj become
Done Raj integrated
```

```python
## Now I will do BOG(bag of words) as LDA works best with it

# creating the dictionary
from gensim.corpora import Dictionary

dicti = Dictionary(df['lemm_with_pos'])

# creating Bag-of-Words Corpus
corpus = [dicti.doc2bow(i) for i in df['lemm_with_pos']]
corpus[:3]
```

```
[[(0, 1), (1, 1), (2, 1), (3, 1), (4, 1), (5, 1), (6, 1)],
 [(7, 1), (8, 1), (9, 1), (10, 1), (11, 1)],
 [(12, 1), (13, 1), (14, 1), (15, 1), (16, 1), (17, 1), (18, 1), (19,
1)]]
```

Task 2

```
# creating the LDA model
from gensim.models import LdaModel

lda_model = LdaModel(corpus=corpus, id2word=dictionary, num_topics=5,
passes=10, random_state=42)
# view topics
topics = lda_model.print_topics(num_words=5)
topics
```

```
[(0,
  '0.045*"attention" + 0.045*"issue" + 0.045*"critical" +
0.045*"change" + 0.045*"global"'),
 (1,
  '0.069*"future" + 0.069*"business" + 0.069*"optimistic" +
0.069*"grow" + 0.069*"economy"'),
 (2,
  '0.045*"uncertainty" + 0.045*"rise" + 0.045*"vehicle" +
0.045*"electric" + 0.045*"volatility"'),
 (3,
  '0.047*"around" + 0.047*"world" + 0.047*"government" +
0.047*"energy" + 0.047*"treatment"'),
 (4,
  '0.068*"industry" + 0.068*"digital" + 0.037*"shift" +
0.037*"concern" + 0.037*"platforms"')]
```

```
import pandas as pd
for topic_num, top in topics:
    words = [i.split('*')[1].replace('"', '').strip() for i in
top.split('+')] # used strip to remove the extra space
    print(f'Topic {topic_num}: {', '.join(words)}')
```

```
Topic 0: attention, issue, critical, change, global
Topic 1: future, business, optimistic, grow, economy
Topic 2: uncertainty, rise, vehicle, electric, volatility
Topic 3: around, world, government, energy, treatment
Topic 4: industry, digital, shift, concern, platforms
```

The LDA model identified five meaningful topics from the document set. The first topic focuses on global challenges and critical issues, highlighted by words such as attention, issue, critical, change, and global. The second topic represents business and economic discussions, emphasizing themes like future, business, optimistic, grow, and economy. The third topic captures market uncertainty and the rise of electric vehicles, as suggested by keywords like uncertainty, rise, vehicle, electric, and volatility. The fourth topic reflects global and governmental developments, with emphasis on world, government, energy, and treatment. Finally, the fifth topic is centered around digital transformation and industry-wide shifts, represented by words such as industry, digital, shift, concern, and platforms. Together, these topics provide an overview of the key themes present in the dataset.

```
!pip install --upgrade pyLDAvis gensim
```

```
Collecting pyLDAvis
  Downloading pyLDAvis-3.4.1-py3-none-any.whl.metadata (4.2 kB)
Requirement already satisfied: gensim in c:\users\lenovo\anaconda3\
lib\site-packages (4.4.0)
Requirement already satisfied: numpy>=1.24.2 in c:\users\lenovo\
anaconda3\lib\site-packages (from pyLDAvis) (2.1.3)
Requirement already satisfied: scipy in c:\users\lenovo\anaconda3\lib\
site-packages (from pyLDAvis) (1.15.3)
Requirement already satisfied: pandas>=2.0.0 in c:\users\lenovo\
anaconda3\lib\site-packages (from pyLDAvis) (2.2.3)
Requirement already satisfied: joblib>=1.2.0 in c:\users\lenovo\
anaconda3\lib\site-packages (from pyLDAvis) (1.4.2)
Requirement already satisfied: jinja2 in c:\users\lenovo\anaconda3\
lib\site-packages (from pyLDAvis) (3.1.6)
Requirement already satisfied: numexpr in c:\users\lenovo\anaconda3\
lib\site-packages (from pyLDAvis) (2.10.1)
Collecting funcy (from pyLDAvis)
  Downloading funcy-2.0-py2.py3-none-any.whl.metadata (5.9 kB)
Requirement already satisfied: scikit-learn>=1.0.0 in c:\users\lenovo\
anaconda3\lib\site-packages (from pyLDAvis) (1.6.1)
Requirement already satisfied: setuptools in c:\users\lenovo\
anaconda3\lib\site-packages (from pyLDAvis) (72.1.0)
Requirement already satisfied: smart_open>=1.8.1 in c:\users\lenovo\
anaconda3\lib\site-packages (from gensim) (7.5.0)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\
lenovo\anaconda3\lib\site-packages (from pandas>=2.0.0->pyLDAvis)
(2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\lenovo\
anaconda3\lib\site-packages (from pandas>=2.0.0->pyLDAvis) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\lenovo\
anaconda3\lib\site-packages (from pandas>=2.0.0->pyLDAvis) (2025.2)
Requirement already satisfied: six>=1.5 in c:\users\lenovo\anaconda3\
lib\site-packages (from python-dateutil>=2.8.2->pandas>=2.0.0-
>pyLDAvis) (1.17.0)
Requirement already satisfied: threadpoolctl>=3.1.0 in c:\users\
lenovo\anaconda3\lib\site-packages (from scikit-learn>=1.0.0-
>pyLDAvis) (3.5.0)
Requirement already satisfied: wrapt in c:\users\lenovo\anaconda3\lib\
site-packages (from smart_open>=1.8.1->gensim) (1.17.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\lenovo\
anaconda3\lib\site-packages (from jinja2->pyLDAvis) (3.0.2)
Downloading pyLDAvis-3.4.1-py3-none-any.whl (2.6 MB)
   ---------------------------------------- 0.0/2.6 MB ? eta -:--:--
   ---------------- ----------------------- 1.0/2.6 MB 10.0 MB/s eta
0:00:01
   ------------------------ --------------- 1.6/2.6 MB 3.4 MB/s eta
0:00:01
   -------------------------------- ------- 2.1/2.6 MB 4.3 MB/s eta
0:00:01
   -------------------------------- ------- 2.1/2.6 MB 4.3 MB/s eta
0:00:01
   -------------------------------- ------- 2.1/2.6 MB 4.3 MB/s eta
0:00:01
   -------------------------------- ------- 2.1/2.6 MB 4.3 MB/s eta
0:00:01
   ---------------------------------------- 2.6/2.6 MB 1.7 MB/s eta
0:00:00
Downloading funcy-2.0-py2.py3-none-any.whl (30 kB)
Installing collected packages: funcy, pyLDAvis

   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   -------------------- ------------------- 1/2 [pyLDAvis]
   ---------------------------------------- 2/2 [pyLDAvis]

Successfully installed funcy-2.0 pyLDAvis-3.4.1
```

```
import pyLDAvis.gensim_models as gensimvis
import pyLDAvis

pyLDAvis.enable_notebook()

vis = gensimvis.prepare(lda_model, corpus, dicti)
vis
```

```
PreparedData(topic_coordinates=              x         y  topics
cluster       Freq
topic
4      0.132299 -0.024336       1        1  28.980018
0     -0.081703 -0.104226       2        1  21.193438
2      0.016342  0.030588       3        1  21.185948
3     -0.061230  0.094314       4        1  19.699370
1     -0.005708  0.003660       5        1   8.941227, topic_info=
Term      Freq     Total Category  logprob  loglift
9         future  1.000000  1.000000  Default  30.0000  30.0000
7       business  0.000000  0.000000  Default  29.0000  29.0000
11    optimistic  0.000000  0.000000  Default  28.0000  28.0000
10          grow  0.000000  0.000000  Default  27.0000  27.0000
8        economy  0.000000  0.000000  Default  26.0000  26.0000
..           ...       ...       ...      ...      ...      ...
23  intelligence  0.067832  1.090495   Topic5  -4.4659  -0.3629
56        become  0.067832  1.090495   Topic5  -4.4659  -0.3629
60       ongoing  0.067832  1.090495   Topic5  -4.4659  -0.3629
42       towards  0.067831  1.090496   Topic5  -4.4659  -0.3629
59    integrated  0.067831  1.090496   Topic5  -4.4659  -0.3629

[247 rows x 6 columns], token_table=      Topic      Freq
Term
term
20        1  0.917015        advance
49        4  0.988103         around
21        1  0.917016     artificial
12        2  0.970871      attention
26        3  0.971029     automobile
56        1  0.917015         become
```

```
13        2  0.970870         change
14        2  0.970870        climate
57        1  0.917014        concern
15        2  0.970870       critical
58        1  0.917014  cybersecurity
37        1  0.594945        digital
0         3  0.971028            due
1         3  0.971028       economic
27        3  0.971028       electric
50        4  0.988103         energy
38        1  0.917015  entertainment
31        4  0.988103         evolve
2         3  0.971028     experience
9         3  0.730523         future
16        2  0.970870         global
51        4  0.988103     government
32        4  0.988103     healthcare
17        2  0.970870      immediate
22        1  0.452380       industry
22        3  0.452380       industry
43        2  0.970872      influence
59        1  0.917014     integrated
23        1  0.917015   intelligence
44        2  0.970871       interact
33        4  0.988103   introduction
52        4  0.988103         invest
18        2  0.970870          issue
3         3  0.971028         market
45        2  0.970871         medium
19        2  0.970871           need
34        4  0.988103            new
60        1  0.917015        ongoing
46        2  0.970871         people
39        1  0.917015       platform
61        1  0.917014      platforms
53        4  0.988103        project
54        4  0.988103      renewable
24        1  0.917015  revolutionize
28        3  0.971028           rise
29        3  0.971028          shape
40        1  0.917013          shift
47        2  0.970871         social
4         3  0.971028          stock
41        1  0.917014      streaming
35        4  0.988103     technology
42        1  0.917014        towards
36        4  0.988103      treatment
5         3  0.971028    uncertainty
30        3  0.971028        vehicle
```

```
6         3  0.971028     volatility
48        2  0.970871            way
55        4  0.988102          world
25        1  0.917015      worldwide, R=30, lambda_step=0.01,
plot_opts={'xlab': 'PC1', 'ylab': 'PC2'}, topic_order=[5, 1, 3, 4, 2])
```

The pyLDAvis visualization provides the proportion of the dataset represented by each topic. According to the visualization, Topic 1 is the most dominant, accounting for 29% of the total content. Topics 2 and 3 contribute equally, each representing 21.2%, indicating that these themes are moderately significant in the dataset. Topic 4 makes up 19.7%, showing a smaller yet meaningful presence. Finally, Topic 5 is the least prominent, covering 8.9% of the text. Together, these proportions sum to approximately 100%, reflecting how the LDA model distributes all textual information across the five identified topics.