Assignment - 1

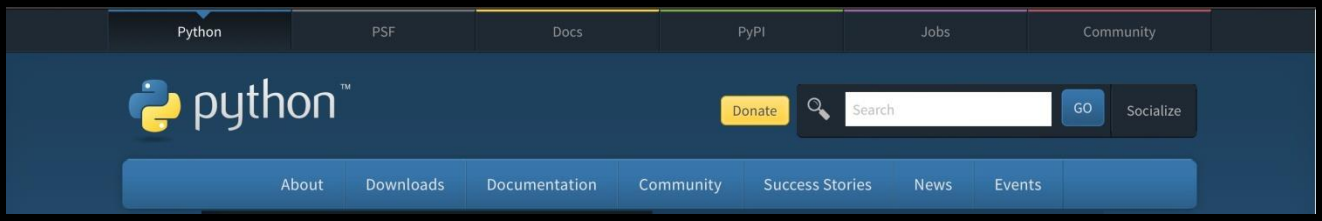
Name: POTHU RAHUL

Batch: 12

Ht.No: 2203A51439

Course: Natural Language Processing (NLP)

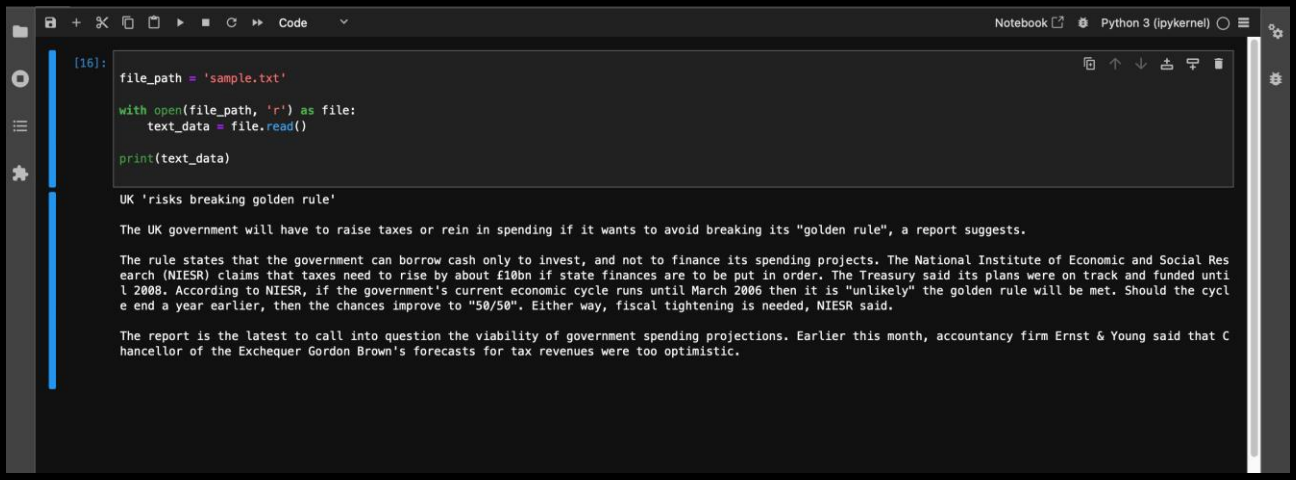
1. Download and install Python from the official website: python.org



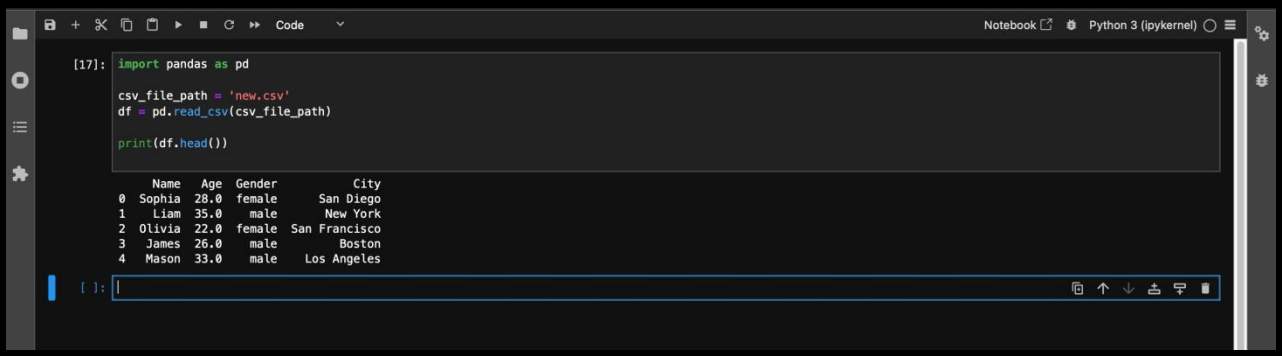

```
C:\Users\HP>python --version
Python 3.12.5
```

2. Loading Text Datasets from Different Resources

(i) Loading a Text File



(ii) Loading Text Data from a CSV File

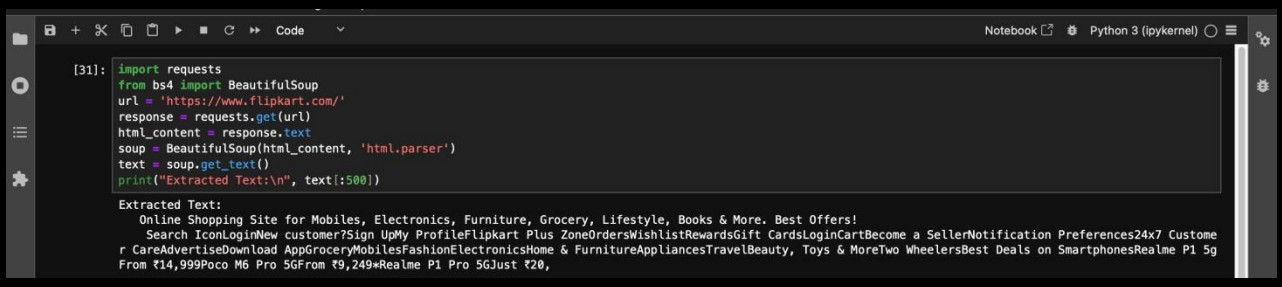


```
[17]: import pandas as pd

      csv_file_path = 'new.csv'
      df = pd.read_csv(csv_file_path)

      print(df.head())
```

```
     Name   Age  Gender           City
0  Sophia  28.0  female      San Diego
1    Liam  35.0    male       New York
2  Olivia  22.0  female  San Francisco
3   James  26.0    male         Boston
4   Mason  33.0    male    Los Angeles
```

```
[ ]:
```

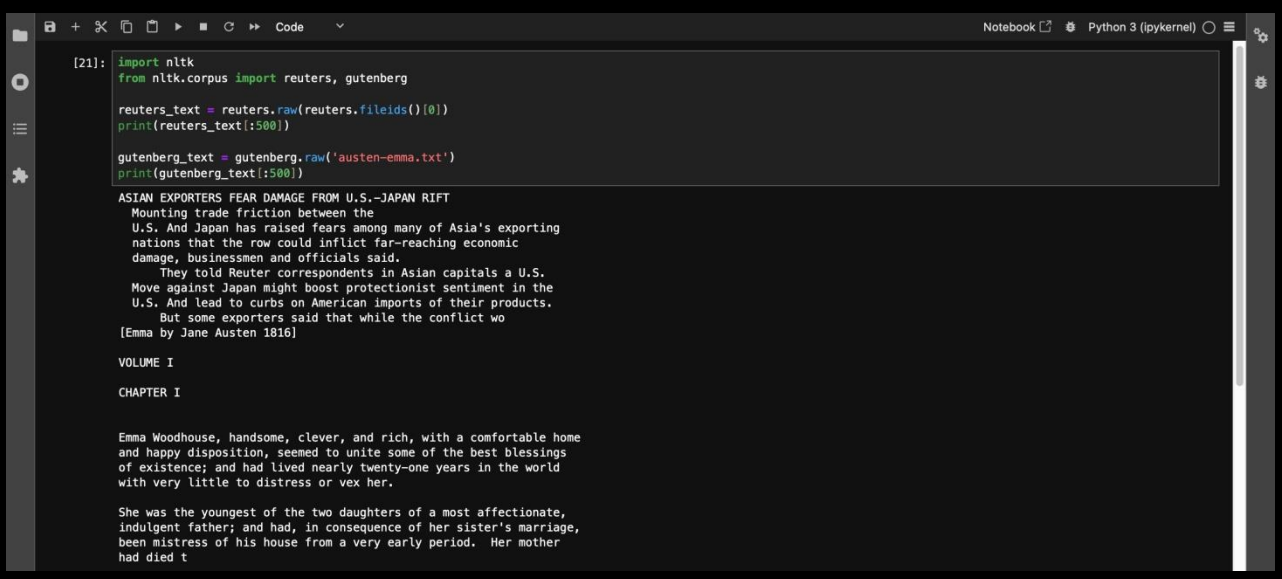
(iii) Loading Text Data from an Online Source



```
[31]: import requests
      from bs4 import BeautifulSoup
      url = 'https://www.flipkart.com/'
      response = requests.get(url)
      html_content = response.text
      soup = BeautifulSoup(html_content, 'html.parser')
      text = soup.get_text()
      print("Extracted Text:\n", text[:500])
```

```
Extracted Text:
   Online Shopping Site for Mobiles, Electronics, Furniture, Grocery, Lifestyle, Books & More. Best Offers!
    Search IconLoginNew customer?Sign UpMy ProfileFlipkart Plus ZoneOrdersWishlistRewardsGift CardsLoginCartBecome a SellerNotification Preferences24x7 Custome
r CareAdvertiseDownload AppGroceryMobilesFashionElectronicsHome & FurnitureAppliancesTravelBeauty, Toys & MoreTwo WheelersBest Deals on SmartphonesRealme P1 5g
From ₹14,999Poco M6 Pro 5GFrom ₹9,249*Realme P1 Pro 5GJust ₹20,
```

(iv) Loading Built-in Text Datasets with NLTK



```
[21]: import nltk
      from nltk.corpus import reuters, gutenberg

      reuters_text = reuters.raw(reuters.fileids()[0])
      print(reuters_text[:500])

      gutenberg_text = gutenberg.raw('austen-emma.txt')
      print(gutenberg_text[:500])
```

```
ASIAN EXPORTERS FEAR DAMAGE FROM U.S.-JAPAN RIFT
  Mounting trade friction between the
  U.S. And Japan has raised fears among many of Asia's exporting
  nations that the row could inflict far-reaching economic
  damage, businessmen and officials said.
      They told Reuter correspondents in Asian capitals a U.S.
  Move against Japan might boost protectionist sentiment in the
  U.S. And lead to curbs on American imports of their products.
      But some exporters said that while the conflict wo
[Emma by Jane Austen 1816]

VOLUME I

CHAPTER I


Emma Woodhouse, handsome, clever, and rich, with a comfortable home
and happy disposition, seemed to unite some of the best blessings
of existence; and had lived nearly twenty-one years in the world
with very little to distress or vex her.

She was the youngest of the two daughters of a most affectionate,
indulgent father; and had, in consequence of her sister's marriage,
been mistress of his house from a very early period.  Her mother
had died t
```

(v) Loading Text Data Using Hugging Face Datasets

```
[28]: from datasets import load_dataset

dataset = load_dataset('ag_news', split='train')

print(dataset[0])
```

```
{'text': "Wall St. Bears Claw Back Into the Black (Reuters) Reuters - Short-sellers, Wall Street's dwindling\\band of ultra-cynics, are seeing green again.", '
label': 2}
```

3. Take your own text or take text as "**The bank can guarantee deposits will eventually cover future tuition costs because it invests in adjustable-rate mortgage securities**." Implement Ambiguity Removal in the text.

```
[9]: import pandas as pd

data = {
    'Text': [
        "The bank can guarantee deposits will eventually cover future tuition costs because it invests in adjustable-rate mortgage securities."
    ]
}

df = pd.DataFrame(data)

print("Original dataset:")
print(df.head())

print("\nMissing values in each column:")
print(df.isnull().sum())



print("\nCleaned data after handling missing values:")
print(df_cleaned.head())

print("\nNumber of duplicate rows:", df_cleaned.duplicated().sum())

df_cleaned = df_cleaned.drop_duplicates()

print("\nData after removing duplicates:")
print(df_cleaned.head())

df_cleaned['Text'] = df_cleaned['Text'].str.lower()

print("\nCleaned data after standardization:")
print(df_cleaned.head())

print("\nFinal cleaned data:")
print(df_cleaned.head())

cleaned_file_path = 'cleaned_sample_dataset.csv'
df_cleaned.to_csv(cleaned_file_path, index=False)
```

Output

```
Original dataset:
                                                Text
0  The bank can guarantee deposits will eventuall...

Missing values in each column:
Text    0
dtype: int64

Cleaned data after handling missing values:
                                                Text
0  the bank can guarantee deposits will eventuall...

Number of duplicate rows: 0

Data after removing duplicates:
                                                Text
0  the bank can guarantee deposits will eventuall...

Cleaned data after standardization:
                                                Text
0  the bank can guarantee deposits will eventuall...

Final cleaned data:
                                                Text
0  the bank can guarantee deposits will eventuall...
```

Output (CSV)

	Text
1	the bank can guarantee deposits will eventually cover future tuition costs because it invests in adjustable-rate mortgage securities.

4. Take your own text or take text as "**Hello there! How are you doing today? NLP is fascinating**." Implement Sentence Segmentation in the text.

```
[15]: import nltk
      from nltk.tokenize import word_tokenize, sent_tokenize

      import pandas as pd

      data = {
          'Text': [
              "Hello there! How are you doing today? NLP is fascinating."
          ]
      }

      df = pd.DataFrame(data)

      df.head()
      print(df.head())
      df['Sentences'] = df['Text'].apply(sent_tokenize)

      df[['Text', 'Sentences']].head()
      df['Words'] = df['Text'].apply(word_tokenize)

      df[['Text', 'Words']].head()
      segmented_file_path = 'segmented_text_dataset.csv'
      df.to_csv(segmented_file_path, index=False)
```

Output

```
                                                Text
0  Hello there! How are you doing today? NLP is f...
```

Output (CSV)

	Text	Sentences	Words
1	Hello there! How are you doing today? NLP is fascinating.	['Hello there!', 'How are you doing today?', 'NLP is fascinating.']	['Hello', 'there', '!', 'How', 'are', 'you', 'doing', 'today', '?', 'NLP', 'is', 'fascinating', '.']