VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

FOUNDATIONS OF DATA SCIENCE

COURSE CODE: BCSE206L

GUIDE NAME: MUKKU NISANTH KARTHEEK

NAME: SIDDHARTH BHARDWAJ

REG NO: 21BCE2001

TITLE: Text Summarization using NLP

INTRODUCTION

Natural Language Processing (NLP) is a branch of artificial intelligence (AI) concerned with the interaction between computers and human language. It involves the development of algorithms and models that enable computers to understand, interpret, and generate human language in a way that is both meaningful and useful. NLP encompasses a wide range of tasks, including text classification, sentiment analysis, machine translation, question answering, and text summarization, among others.

Text summarization is the process of condensing a longer piece of text, such as an article, document, or book, into a shorter version while retaining its key ideas and main points. The goal of text summarization is to provide a concise overview of the original text that captures its essence and important information.

There are generally two approaches to text summarization:

1. **Extractive Summarization**: In extractive summarization, the summary is generated by selecting and extracting the most important sentences or phrases from the original text. These sentences are chosen based on criteria such as their relevance to the main topic, importance, and informativeness. Extractive summarization methods often involve ranking sentences using features like word frequency, sentence length, and similarity to other sentences in the text.

2. **Abstractive Summarization**: Abstractive summarization goes beyond simply selecting and extracting sentences from the original text. Instead, it involves generating new sentences that capture the meaning and essence of the source text in a more concise form. Abstractive summarization methods use natural language processing techniques to understand the content of the text and generate summaries that may contain paraphrased or rephrased versions of the original sentences.

In this project, we will implement the summarisation of articles that are extracted from datasets available online.

MOTIVATION/ APPLICATIONS

Undertaking a project on text summarization is valuable because it helps people deal with too much information. Instead of reading long articles or documents, they can quickly understand the main points through summaries. This is useful for staying updated on news, finding important research in academic work, and managing content in businesses. Text summarization makes it easier to find key information, saving time and effort, and it's helpful in many areas where dealing with lots of text is common. A text summarization project offers practical benefits in information management, decision support, and knowledge dissemination across various fields and industries.

Some practical applications of Text Summarisation are:

1. **Information Overload Management**: With the exponential growth of digital content, individuals and organizations are saturated with vast amounts of textual information. Text summarization helps manage this overload by condensing lengthy documents or articles into concise summaries, allowing users to quickly extract relevant information without having to read through entire texts.

2. **Time Efficiency**: Text summarization enables users to save time by providing them with quick overviews of lengthy documents or articles, allowing them to extract key insights and make informed decisions more efficiently.

3. **Enhanced Accessibility**: Summarized content is often more accessible to a wider audience, including individuals with limited time or attention spans, non-native speakers, or those with disabilities that may affect reading comprehension. By providing concise summaries, text summarization tools can make information more accessible and inclusive.

4. **Improved Information Retrieval**: Summarized content can serve as valuable metadata for information retrieval systems, helping users quickly assess the relevance of documents or articles before delving into them in more detail. This can improve search efficiency and user satisfaction with information retrieval platforms.

LEARNINGS

By doing this project: text summarization project using NLP in Python, with libraries like NumPy and Pandas, on Kaggle platform, I've gained several valuable skills and insights.

1. I learned the basics of Natural Language Processing (NLP), including text preprocessing, feature extraction, and understanding various NLP techniques such as tokenization, stemming, and lemmatization.

2. I gained hands-on experience in handling text data using Python, including techniques like data cleaning, tokenization, and vectorization.

3. Explored different text summarization algorithms, both extractive and abstractive, which provided insights into how they work under the hood.

4. Applied machine learning models to text data for tasks like summarization, deepening my understanding of NLP and machine learning.

5. Became familiar with data manipulation and analysis using the pandas and NumPy libraries, including tasks like loading, cleaning, and transforming text data.

6. Evaluated the performance of text summarization models using appropriate evaluation metrics and techniques, learning how to fine-tune and optimize models for better results.

7. Worked on Kaggle, gaining hands-on experience with real-world datasets, competing with others, and learning from the broader data science community through discussions, kernels, and competitions.

Overall, the project provided a learning experience in NLP, machine learning and data analysis.

CODE:

1. Importing the required Libraries

```
import numpy as np import pandas as
pd import warnings import re import
nltk from nltk import word_tokenize
from nltk.tokenize import sent_tokenize
from textblob import TextBlob import
string from string import punctuation
from nltk.corpus import stopwords
from statistics import mean from heapq
import nlargest from wordcloud import
WordCloud import seaborn as sns
import matplotlib.pyplot as plt

stop_words = set(stopwords.words('english')) punctuation =
punctuation + '\n' + '—' + '“' + ',' + '”' + '‘' + '-' + '’'
warnings.filterwarnings('ignore')

# Importing the dataset df_1 = pd.read_csv("/kaggle/input/all-
the-news/articles1.csv") df_2 =
pd.read_csv("/kaggle/input/all-the-news/articles2.csv") df_3 =
pd.read_csv("/kaggle/input/all-the-news/articles3.csv")

# Making one Dataframe by appending all datasets
d = [df_1, df_2, df_3] df =
pd.concat(d, keys = ['x', 'y', 'z'])
df.rename(columns = {'content' : 'article'}, inplace = True);

# Shape of the dataset df.shape

# Drop unnecessary columns df.drop(columns =
['Unnamed: 0'], inplace = True) df.head()
```

2. Making the Article Summarizer

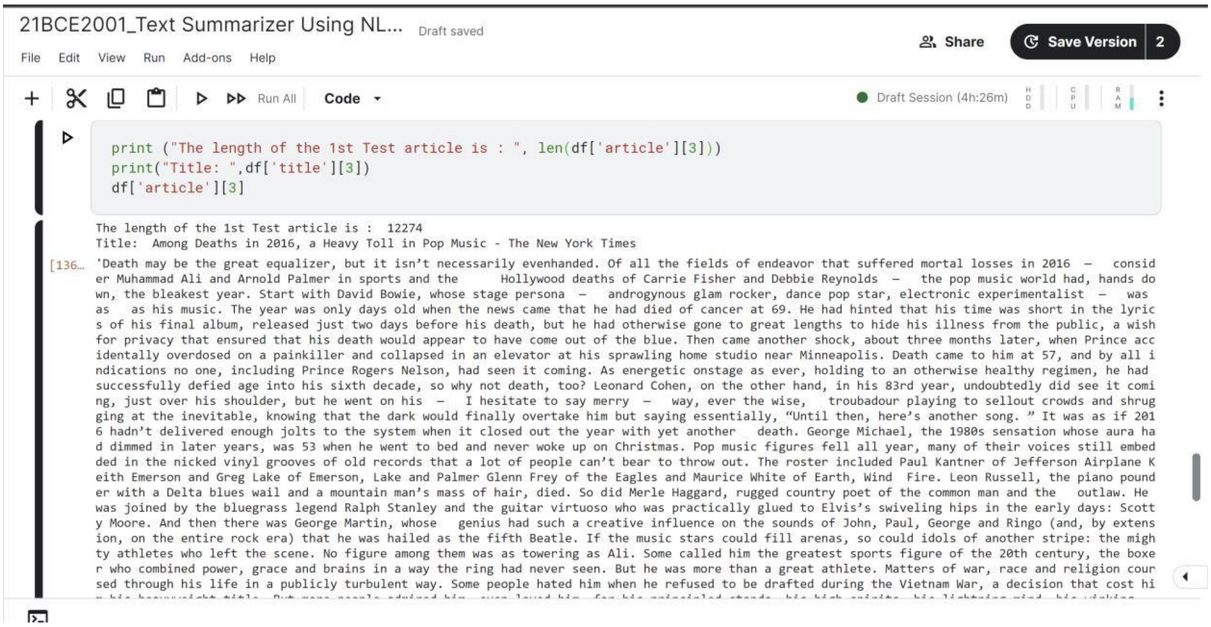
```
contractions_dict = {  "ain't":
"am not",
"aren't": "are not",
"can't": "cannot",
"can't've": "cannot have",
"'cause": "because",
"could've": "could have",
"couldn't": "could not",
"couldn't've": "could not have",
"didn't": "did not",
"doesn't": "does not", "doesn’t":
"does not",
"don't": "do not", "don’t":
"do not",
"hadn't": "had not",
"hadn't've": "had not have",
"hasn't": "has not",
"haven't": "have not",
"he'd": "he had",
"he'd've": "he would have",
"he'll": "he will",
"he'll've": "he will have",
"he's": "he is",
"how'd": "how did",
"how'd'y": "how do you",
"how'll": "how will",
"how's": "how is",
"i'd": "i would",
"i'd've": "i would have",
"i'll": "i will",
"i'll've": "i will have",
"i'm": "i am",
"i've": "i have",
"isn't": "is not",
"it'd": "it would",
"it'd've": "it would have",
"it'll": "it will",
"it'll've": "it will have",
"it's": "it is",
"let's": "let us",
"ma'am": "madam",
"mayn't": "may not",
"might've": "might have",
"mightn't": "might not",
"mightn't've": "might not have",
"must've": "must have",
"mustn't": "must not",
"mustn't've": "must not have",
"needn't": "need not",
"needn't've": "need not have",
"o'clock": "of the clock", "oughtn't":
"ought not",
"oughtn't've": "ought not have",
"shan't": "shall not",
"sha'n't": "shall not",
"shan't've": "shall not have",
"she'd": "she would",
"she'd've": "she would have",
"she'll": "she will",
"she'll've": "she will have",
"she's": "she is",
"should've": "should have",
"shouldn't": "should not",
"shouldn't've": "should not have",
"so've": "so have",
"so's": "so is",
"that'd": "that would",
"that'd've": "that would have",
"that's": "that is",
"there'd": "there would",
"there'd've": "there would have",
"there's": "there is",
"they'd": "they would",
"they'd've": "they would have",
"they'll": "they will",
"they'll've": "they will have",
"they're": "they are",
"they've": "they have",
"to've": "to have",
"wasn't": "was not", "we'd":
"we would",
"we'd've": "we would have",
"we'll": "we will",
"we'll've": "we will have",
"we're": "we are",
"we've": "we have",
"weren't": "were not",
"what'll": "what will",
"what'll've": "what will have",
"what're": "what are",
"what's": "what is",
"what've": "what have",
"when's": "when is",
"when've": "when have",
"where'd": "where did",
"where's": "where is",
"where've": "where have",
"who'll": "who will",
"who'll've": "who will have",
"who's": "who is",
"who've": "who have",
"why's": "why is",
"why've": "why have",
"will've": "will have",
"won't": "will not",
"won't've": "will not have",
"would've": "would have",
"wouldn't": "would not",
"wouldn't've": "would not have",
"y'all": "you all", "y’all":
"you all",
"y'all'd": "you all would",
"y'all'd've": "you all would have",
"y'all're": "you all are",
"y'all've": "you all have",
"you'd": "you would",
"you'd've": "you would have",
"you'll": "you will",
"you'll've": "you will have",
"you're": "you are",
"you've": "you have",
"ain’t": "am not",
"aren’t": "are not",
"can’t": "cannot",
"can’t’ve": "cannot have",
"’cause": "because",
"could’ve": "could have",
"couldn’t": "could not",
"couldn’t’ve": "could not have",
"didn’t": "did not",
"doesn’t": "does not",
"don’t": "do not",
"don’t": "do not",
"hadn’t": "had not",
"hadn’t’ve": "had not have",
"hasn’t": "has not",
"haven’t": "have not",
"he’d": "he had",
"he’d’ve": "he would have",
"he’ll": "he will",
"he’ll’ve": "he will have",
"he’s": "he is",
"how’d": "how did",
"how’d’y": "how do you",
"how’ll": "how will",
"how’s": "how is",
"i’d": "i would",
"i’d’ve": "i would have",
"i’ll": "i will",
"i’ll’ve": "i will have",
"i’m": "i am",
"i’ve": "i have",
"isn’t": "is not",
"it’d": "it would",
"it’d’ve": "it would have",
"it’ll": "it will",
"it’ll’ve": "it will have",
"it’s": "it is",
"let’s": "let us",
"ma’am": "madam",
"mayn’t": "may not",
"might’ve": "might have",
"mightn’t": "might not",
"mightn’t’ve": "might not have",
"must’ve": "must have",
"mustn’t": "must not",
"mustn’t’ve": "must not have", "needn’t":
"need not",
"needn’t’ve": "need not have",
"o’clock": "of the clock", "oughtn’t":
"ought not",
"oughtn’t’ve": "ought not have",
"shan’t": "shall not",
"sha’n’t": "shall not",
"shan’t’ve": "shall not have",
"she’d": "she would",
"she’d’ve": "she would have",
"she’ll": "she will",
"she’ll’ve": "she will have",
"she’s": "she is",
"should’ve": "should have",
"shouldn’t": "should not",
"shouldn’t’ve": "should not have",
"so’ve": "so have",
"so’s": "so is",
"that’d": "that would",
"that’d’ve": "that would have",
"that’s": "that is",
"there’d": "there would",
"there’d’ve": "there would have",
"there’s": "there is",
"they’d": "they would",
"they’d’ve": "they would have",
"they’ll": "they will",
"they’ll’ve": "they will have",
"they’re": "they are",
"they’ve": "they have",
"to’ve": "to have",
"wasn’t": "was not", "we’d":
"we would",
"we’d’ve": "we would have",
"we’ll": "we will",
"we’ll’ve": "we will have",
"we’re": "we are",
"we’ve": "we have",
"weren’t": "were not",
"what’ll": "what will",
"what’ll’ve": "what will have",
"what’re": "what are",
"what’s": "what is",
"what’ve": "what have",
"when’s": "when is",
"when’ve": "when have",
"where’d": "where did",
"where’s": "where is",
"where’ve": "where have",
"who’ll": "who will",
"who’ll’ve": "who will have",
"who’s": "who is",
"who’ve": "who have",
"why’s": "why is",
"why’ve": "why have",
"will’ve": "will have",
"won’t": "will not",
"won’t’ve": "will not have",
"would’ve": "would have",
"wouldn’t": "would not",
"wouldn’t’ve": "would not have",
"y’all": "you all",
"y’all": "you all",
"y’all’d": "you all would",
"y’all’d’ve": "you all would have",
"y’all’re": "you all are",
"y’all’ve": "you all have",
"you’d": "you would",
"you’d’ve": "you would have",
"you’ll": "you will",
"you’ll’ve": "you will have",
"you’re": "you are",
"you’re": "you are",
"you’ve": "you have",
}
contractions_re = re.compile('(%s)' % '|'.join(contractions_dict.keys()))
# Function to clean the html from the article
def cleanhtml(raw_html):     cleanr =
re.compile('<.*?>')     cleantext =
re.sub(cleanr, '', raw_html)     return
cleantext

# Function expand the contractions if there's any def
expand_contractions(s, contractions_dict=contractions_dict):
def replace(match):         return
contractions_dict[match.group(0)]
    return contractions_re.sub(replace, s)

# Function to preprocess the articles def
preprocessing(article):
    global article_sent

    # Converting to lowercase
    article = article.str.lower()

    # Removing the HTML
    article = article.apply(lambda x: cleanhtml(x))

    # Removing the email ids
    article = article.apply(lambda x: re.sub('\S+@\S+','', x))

    # Removing The URLS
    article = article.apply(lambda x: re.sub("((http\://|https\://|ftp\://)|(www.))+(([a-zA-Z0-9\.-
]+\.[azA-Z]{2,4})|([0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}))(/[a-zA-Z0-9%:/-_\?\.'~]*)?",'', x))

    # Removing the '\xa0'
    article = article.apply(lambda x: x.replace("\xa0", " "))

    # Removing the contractions
    article = article.apply(lambda x: expand_contractions(x))

    # Stripping the possessives     article =
article.apply(lambda x: x.replace("'s", ''))     article =
article.apply(lambda x: x.replace(’s', ''))     article =
article.apply(lambda x: x.replace("\'s", ''))     article =
article.apply(lambda x: x.replace("\’s", ''))

    # Removing the Trailing and leading whitespace and double spaces
article = article.apply(lambda x: re.sub(' +', ' ',x))
```

```
   # Copying the article for the sentence tokenization
article_sent = article.copy()

   # Removing punctuations from the article
   article = article.apply(lambda x: ''.join(word for word in x if word not in punctuation))

   # Removing the Trailing and leading whitespace and double spaces again as removing punctuation
might
   # Lead to a white space
   article = article.apply(lambda x: re.sub(' +', ' ',x))

   # Removing the Stopwords
   article = article.apply(lambda x: ' '.join(word for word in x.split() if word not in stop_words))

   return article

# Function to normalize the word frequency which is used in the function word_frequency
def normalize(li_word):     global
normalized_freq     normalized_freq = []     for
dictionary in li_word:         max_frequency =
max(dictionary.values())        for word in
dictionary.keys():
          dictionary[word] = dictionary[word]/max_frequency
normalized_freq.append(dictionary)     return
normalized_freq

# Function to calculate the word frequency def
word_frequency(article_word):
   word_frequency = {}     li_word = []     for
sentence in article_word:         for word in
word_tokenize(sentence):            if word
not in word_frequency.keys():
word_frequency[word] = 1            else:
             word_frequency[word] += 1
li_word.append(word_frequency)
      word_frequency = {}
normalize(li_word)
   return normalized_freq

# Function to Score the sentence which is called in the function sent_token
def sentence_score(li):     global sentence_score_list     sentence_score = {}
sentence_score_list = []     for list_, dictionary in zip(li, normalized_freq):
for sent in list_:
          for word in word_tokenize(sent):
if word in dictionary.keys():
                 if sent not in sentence_score.keys():
                    sentence_score[sent] = dictionary[word]
else:
```

```
                  sentence_score[sent] += dictionary[word]
sentence_score_list.append(sentence_score)
      sentence_score = {}
   return sentence_score_list

# Function to tokenize the sentence def
sent_token(article_sent):
   sentence_list = []
sent_token = []     for
sent in article_sent:
      token = sent_tokenize(sent)        for sentence in token:           token_2 =
''.join(word for word in sentence if word not in punctuation)           token_2 =
re.sub(' +', ' ',token_2)
         sent_token.append(token_2)
sentence_list.append(sent_token)
      sent_token = []
sentence_score(sentence_list)     return
sentence_score_list

# Function which generates the summary of the articles (This uses the 20% of the sentences with the
highest score) def summary(sentence_score_OwO):     summary_list = []     for summ in
sentence_score_OwO:        select_length = int(len(summ)*0.25)
      summary_ = nlargest(select_length, summ, key = summ.get)
summary_list.append(".".join(summary_))     return
summary_list


# Functions to change the article string (if passed) to change it to generate a pandas series
def make_series(art):     global dataframe     data_dict = {'article' : [art]}
   dataframe = pd.DataFrame(data_dict)['article']
return dataframe

# Function which is to be called to generate the summary which in further calls other functions
alltogether def article_summarize(artefact):

   if type(artefact) != pd.Series:
      artefact = make_series(artefact)

   df = preprocessing(artefact)

   word_normalization = word_frequency(df)

   sentence_score_OwO = sent_token(article_sent)

   summarized_article = summary(sentence_score_OwO)

   return summarized_article
```

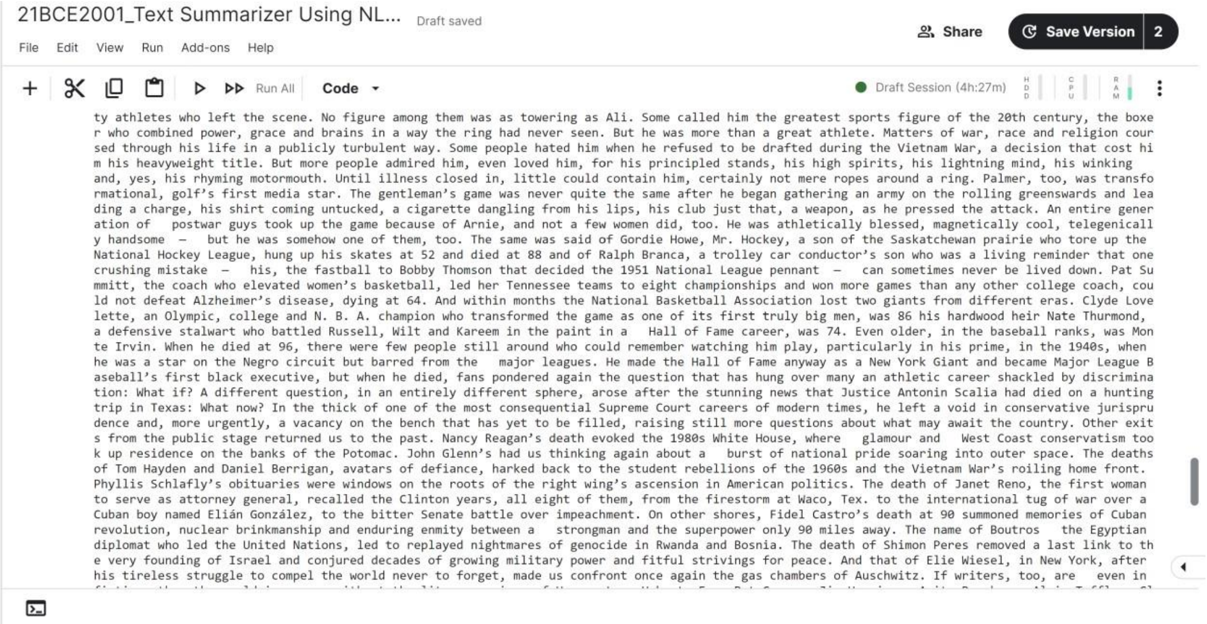
```
# Generating the Word Cloud of the article using the preprocessing and make_series function
mentioned below def word_cloud(art):
   art_ = make_series(art)
OwO = preprocessing(art_)
   wordcloud_ = WordCloud(height = 500, width = 1000, background_color = 'white').generate(art)
plt.figure(figsize=(15, 10))
   plt.imshow(wordcloud_, interpolation='bilinear')
plt.axis('off');
#summaries for the first 5 articles
summaries = article_summarize(df['article'][0:5])

print ("The length of the 1st Test article is : ", len(df['article'][3]))
print("Title: ",df['title'][3]) df['article'][3]

print ("The length of the summarized article 1 is : ", len(summaries[3]))
print("Title: ",df['title'][3]) summaries[3]

print ("The length of the 2nd Test article is : ", len(df['article'][4]))
print("Title: ",df['title'][4]) df['article'][4]

print ("The length of the summarized article 2 is : ", len(summaries[4]))
print("Title: ",df['title'][4]) summaries[4]
```

SAMPLE INPUT 1:



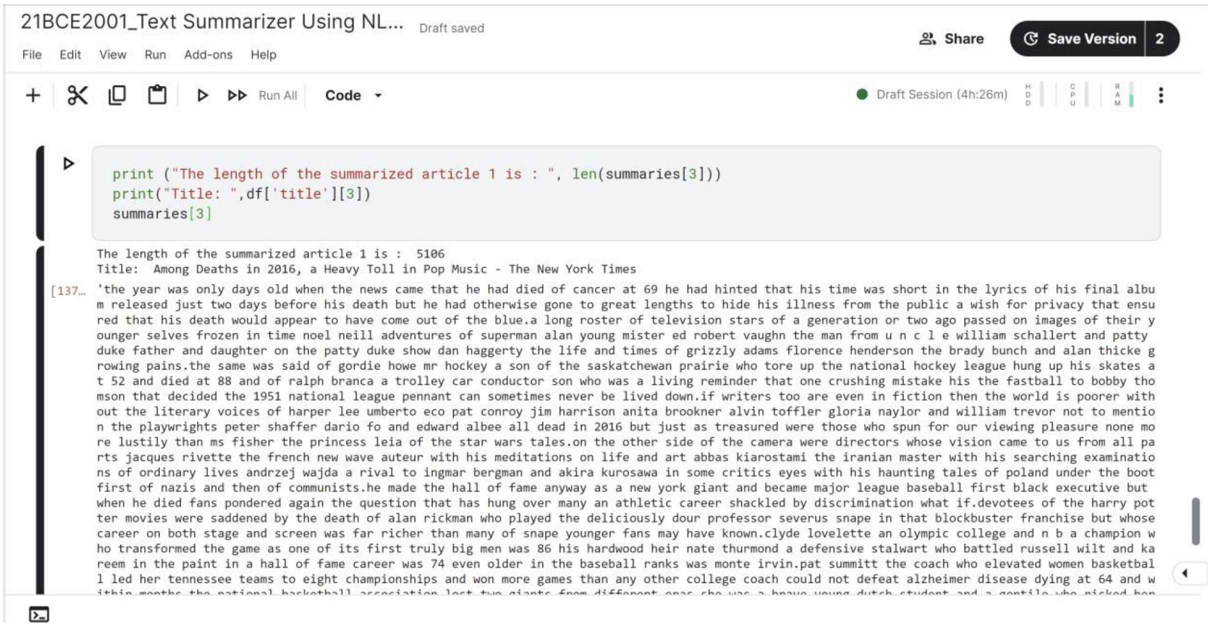
```
print ("The length of the 1st Test article is : ", len(df['article'][3]))
print("Title: ",df['title'][3])
df['article'][3]
```

```
The length of the 1st Test article is :  12274
Title:  Among Deaths in 2016, a Heavy Toll in Pop Music - The New York Times
[136… 'Death may be the great equalizer, but it isn’t necessarily evenhanded. Of all the fields of endeavor that suffered mortal losses in 2016 —   consid
er Muhammad Ali and Arnold Palmer in sports and the      Hollywood deaths of Carrie Fisher and Debbie Reynolds —   the pop music world had, hands do
wn, the bleakest year. Start with David Bowie, whose stage persona —   androgynous glam rocker, dance pop star, electronic experimentalist —   was
as   as his music. The year was only days old when the news came that he had died of cancer at 69. He had hinted that his time was short in the lyric
s of his final album, released just two days before his death, but he had otherwise gone to great lengths to hide his illness from the public, a wish
for privacy that ensured that his death would appear to have come out of the blue. Then came another shock, about three months later, when Prince acc
identally overdosed on a painkiller and collapsed in an elevator at his sprawling home studio near Minneapolis. Death came to him at 57, and by all i
ndications no one, including Prince Rogers Nelson, had seen it coming. As energetic onstage as ever, holding to an otherwise healthy regimen, he had
successfully defied age into his sixth decade, so why not death, too? Leonard Cohen, on the other hand, in his 83rd year, undoubtedly did see it comi
ng, just over his shoulder, but he went on his —   I hesitate to say merry —   way, ever the wise,   troubadour playing to sellout crowds and shrug
ging at the inevitable, knowing that the dark would finally overtake him but saying essentially, “Until then, here’s another song. ” It was as if 201
6 hadn’t delivered enough jolts to the system when it closed out the year with yet another   death. George Michael, the 1980s sensation whose aura ha
d dimmed in later years, was 53 when he went to bed and never woke up on Christmas. Pop music figures fell all year, many of their voices still embed
ded in the nicked vinyl grooves of old records that a lot of people can’t bear to throw out. The roster included Paul Kantner of Jefferson Airplane K
eith Emerson and Greg Lake of Emerson, Lake and Palmer Glenn Frey of the Eagles and Maurice White of Earth, Wind  Fire. Leon Russell, the piano pound
er with a Delta blues wail and a mountain man’s mass of hair, died. So did Merle Haggard, rugged country poet of the common man and the   outlaw. He
was joined by the bluegrass legend Ralph Stanley and the guitar virtuoso who was practically glued to Elvis’s swiveling hips in the early days: Scott
y Moore. And then there was George Martin, whose   genius had such a creative influence on the sounds of John, Paul, George and Ringo (and, by extens
ion, on the entire rock era) that he was hailed as the fifth Beatle. If the music stars could fill arenas, so could idols of another stripe: the migh
ty athletes who left the scene. No figure among them was as towering as Ali. Some called him the greatest sports figure of the 20th century, the boxe
r who combined power, grace and brains in a way the ring had never seen. But he was more than a great athlete. Matters of war, race and religion cour
sed through his life in a publicly turbulent way. Some people hated him when he refused to be drafted during the Vietnam War, a decision that cost hi
```



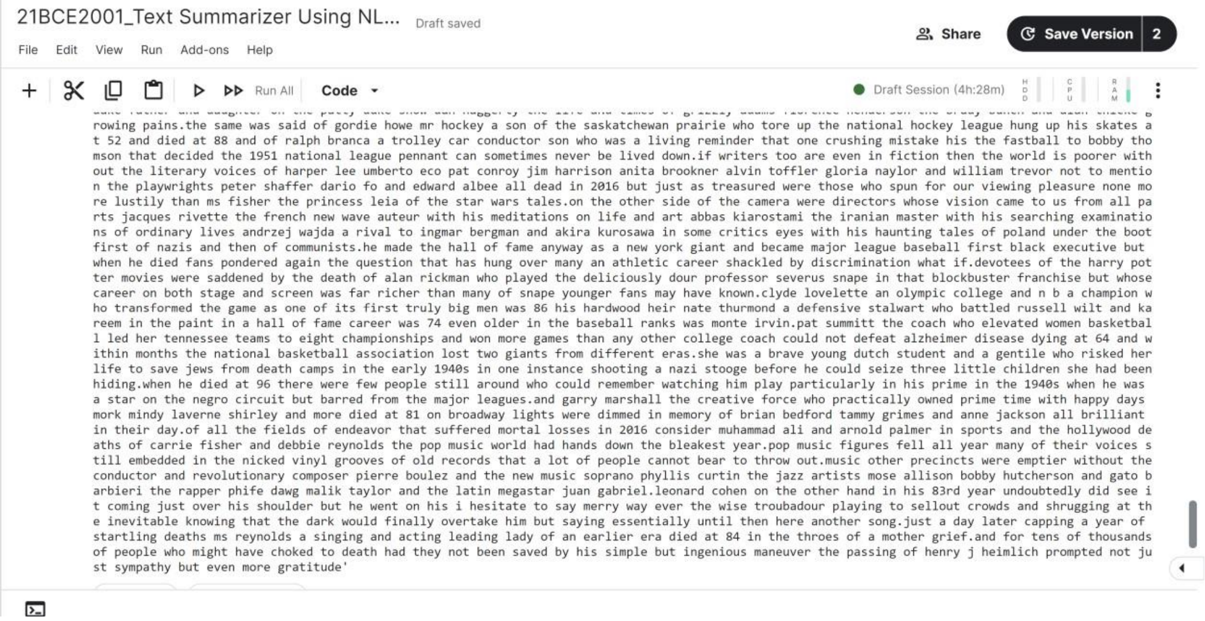
```
ty athletes who left the scene. No figure among them was as towering as Ali. Some called him the greatest sports figure of the 20th century, the boxe
r who combined power, grace and brains in a way the ring had never seen. But he was more than a great athlete. Matters of war, race and religion cour
sed through his life in a publicly turbulent way. Some people hated him when he refused to be drafted during the Vietnam War, a decision that cost hi
m his heavyweight title. But more people admired him, even loved him, for his principled stands, his high spirits, his lightning mind, his winking
and, yes, his rhyming motormouth. Until illness closed in, little could contain him, certainly not mere ropes around a ring. Palmer, too, was transfo
rmational, golf’s first media star. The gentleman’s game was never quite the same after he began gathering an army on the rolling greenswards and lea
ding a charge, his shirt coming untucked, a cigarette dangling from his lips, his club just that, a weapon, as he pressed the attack. An entire gener
ation of   postwar guys took up the game because of Arnie, and not a few women did, too. He was athletically blessed, magnetically cool, telegenicall
y handsome —   but he was somehow one of them, too. The same was said of Gordie Howe, Mr. Hockey, a son of the Saskatchewan prairie who tore up the
National Hockey League, hung up his skates at 52 and died at 88 and of Ralph Branca, a trolley car conductor’s son who was a living reminder that one
crushing mistake —   his, the fastball to Bobby Thomson that decided the 1951 National League pennant —   can sometimes never be lived down. Pat Su
mmitt, the coach who elevated women’s basketball, led her Tennessee teams to eight championships and won more games than any other college coach, cou
ld not defeat Alzheimer’s disease, dying at 64. And within months the National Basketball Association lost two giants from different eras. Clyde Love
lette, an Olympic, college and N. B. A. champion who transformed the game as one of its first truly big men, was 86 his hardwood heir Nate Thurmond,
a defensive stalwart who battled Russell, Wilt and Kareem in the paint in a   Hall of Fame career, was 74. Even older, in the baseball ranks, was Mon
te Irvin. When he died at 96, there were few people still around who could remember watching him play, particularly in his prime, in the 1940s, when
he was a star on the Negro circuit but barred from the   major leagues. He made the Hall of Fame anyway as a New York Giant and became Major League B
aseball’s first black executive, but when he died, fans pondered again the question that has hung over many an athletic career shackled by discrimina
tion: What if? A different question, in an entirely different sphere, arose after the stunning news that Justice Antonin Scalia had died on a hunting
trip in Texas: What now? In the thick of one of the most consequential Supreme Court careers of modern times, he left a void in conservative jurispru
dence and, more urgently, a vacancy on the bench that has yet to be filled, raising still more questions about what may await the country. Other exit
s from the public stage returned us to the past. Nancy Reagan’s death evoked the 1980s White House, where   glamour and   West Coast conservatism too
k up residence on the banks of the Potomac. John Glenn’s had us thinking again about a   burst of national pride soaring into outer space. The deaths
of Tom Hayden and Daniel Berrigan, avatars of defiance, harked back to the student rebellions of the 1960s and the Vietnam War’s roiling home front.
Phyllis Schlafly’s obituaries were windows on the roots of the right wing’s ascension in American politics. The death of Janet Reno, the first woman
to serve as attorney general, recalled the Clinton years, all eight of them, from the firestorm at Waco, Tex. to the international tug of war over a
Cuban boy named Elián González, to the bitter Senate battle over impeachment. On other shores, Fidel Castro’s death at 90 summoned memories of Cuban
revolution, nuclear brinkmanship and enduring enmity between a   strongman and the superpower only 90 miles away. The name of Boutros   the Egyptian
diplomat who led the United Nations, led to replayed nightmares of genocide in Rwanda and Bosnia. The death of Shimon Peres removed a last link to th
e very founding of Israel and conjured decades of growing military power and fitful strivings for peace. And that of Elie Wiesel, in New York, after
his tireless struggle to compel the world never to forget, made us confront once again the gas chambers of Auschwitz. If writers, too, are   even in
```

OUTPUT 1:



```
print ("The length of the summarized article 1 is : ", len(summaries[3]))
print("Title: ",df['title'][3])
summaries[3]
```

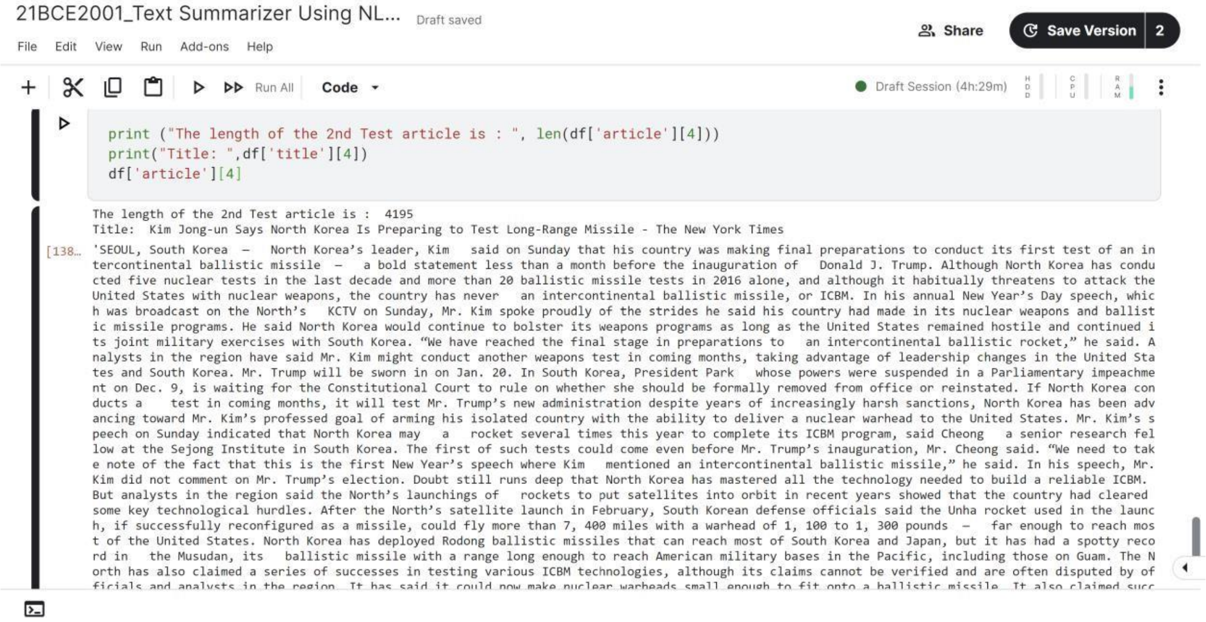
```
The length of the summarized article 1 is :  5106
Title:  Among Deaths in 2016, a Heavy Toll in Pop Music - The New York Times
'the year was only days old when the news came that he had died of cancer at 69 he had hinted that his time was short in the lyrics of his final albu
m released just two days before his death but he had otherwise gone to great lengths to hide his illness from the public a wish for privacy that ensu
red that his death would appear to have come out of the blue.a long roster of television stars of a generation or two ago passed on images of their y
ounger selves frozen in time noel neill adventures of superman alan young mister ed robert vaughn the man from u n c l e william schallert and patty
duke father and daughter on the patty duke show dan haggerty the life and times of grizzly adams florence henderson the brady bunch and alan thicke g
rowing pains.the same was said of gordie howe mr hockey a son of the saskatchewan prairie who tore up the national hockey league hung up his skates a
t 52 and died at 88 and of ralph branca a trolley car conductor son who was a living reminder that one crushing mistake his the fastball to bobby tho
mson that decided the 1951 national league pennant can sometimes never be lived down.if writers too are even in fiction then the world is poorer with
out the literary voices of harper lee umberto eco pat conroy jim harrison anita brookner alvin toffler gloria naylor and william trevor not to mentio
n the playwrights peter shaffer dario fo and edward albee all dead in 2016 but just as treasured were those who spun for our viewing pleasure none mo
re lustily than ms fisher the princess leia of the star wars tales.on the other side of the camera were directors whose vision came to us from all pa
rts jacques rivette the french new wave auteur with his meditations on life and art abbas kiarostami the iranian master with his searching examinatio
ns of ordinary lives andrzej wajda a rival to ingmar bergman and akira kurosawa in some critics eyes with his haunting tales of poland under the boot
first of nazis and then of communists.he made the hall of fame anyway as a new york giant and became major league baseball first black executive but
when he died fans pondered again the question that has hung over many an athletic career shackled by discrimination what if.devotees of the harry pot
ter movies were saddened by the death of alan rickman who played the deliciously dour professor severus snape in that blockbuster franchise but whose
career on both stage and screen was far richer than many of snape younger fans may have known.clyde lovelette an olympic college and n b a champion w
ho transformed the game as one of its first truly big men was 86 his hardwood heir nate thurmond a defensive stalwart who battled russell wilt and ka
reem in the paint in a hall of fame career was 74 even older in the baseball ranks was monte irvin.pat summitt the coach who elevated women basketbal
l led her tennessee teams to eight championships and won more games than any other college coach could not defeat alzheimer disease dying at 64 and w
```



```
rowing pains.the same was said of gordie howe mr hockey a son of the saskatchewan prairie who tore up the national hockey league hung up his skates a
t 52 and died at 88 and of ralph branca a trolley car conductor son who was a living reminder that one crushing mistake his the fastball to bobby tho
mson that decided the 1951 national league pennant can sometimes never be lived down.if writers too are even in fiction then the world is poorer with
out the literary voices of harper lee umberto eco pat conroy jim harrison anita brookner alvin toffler gloria naylor and william trevor not to mentio
n the playwrights peter shaffer dario fo and edward albee all dead in 2016 but just as treasured were those who spun for our viewing pleasure none mo
re lustily than ms fisher the princess leia of the star wars tales.on the other side of the camera were directors whose vision came to us from all pa
rts jacques rivette the french new wave auteur with his meditations on life and art abbas kiarostami the iranian master with his searching examinatio
ns of ordinary lives andrzej wajda a rival to ingmar bergman and akira kurosawa in some critics eyes with his haunting tales of poland under the boot
first of nazis and then of communists.he made the hall of fame anyway as a new york giant and became major league baseball first black executive but
when he died fans pondered again the question that has hung over many an athletic career shackled by discrimination what if.devotees of the harry pot
ter movies were saddened by the death of alan rickman who played the deliciously dour professor severus snape in that blockbuster franchise but whose
career on both stage and screen was far richer than many of snape younger fans may have known.clyde lovelette an olympic college and n b a champion w
ho transformed the game as one of its first truly big men was 86 his hardwood heir nate thurmond a defensive stalwart who battled russell wilt and ka
reem in the paint in a hall of fame career was 74 even older in the baseball ranks was monte irvin.pat summitt the coach who elevated women basketbal
l led her tennessee teams to eight championships and won more games than any other college coach could not defeat alzheimer disease dying at 64 and w
```

```
ithin months the national basketball association lost two giants from different eras.she was a brave young dutch student and a gentile who risked her
life to save jews from death camps in the early 1940s in one instance shooting a nazi stooge before he could seize three little children she had been
hiding.when he died at 96 there were few people still around who could remember watching him play particularly in his prime in the 1940s when he was
a star on the negro circuit but barred from the major leagues.and garry marshall the creative force who practically owned prime time with happy days
mork mindy laverne shirley and more died at 81 on broadway lights were dimmed in memory of brian bedford tammy grimes and anne jackson all brilliant
in their day.of all the fields of endeavor that suffered mortal losses in 2016 consider muhammad ali and arnold palmer in sports and the hollywood de
aths of carrie fisher and debbie reynolds the pop music world had hands down the bleakest year.pop music figures fell all year many of their voices s
till embedded in the nicked vinyl grooves of old records that a lot of people cannot bear to throw out.music other precincts were emptier without the
conductor and revolutionary composer pierre boulez and the new music soprano phyllis curtin the jazz artists mose allison bobby hutcherson and gato b
arbieri the rapper phife dawg malik taylor and the latin megastar juan gabriel.leonard cohen on the other hand in his 83rd year undoubtedly did see i
t coming just over his shoulder but he went on his i hesitate to say merry way ever the wise troubadour playing to sellout crowds and shrugging at th
e inevitable knowing that the dark would finally overtake him but saying essentially until then here another song.just a day later capping a year of
startling deaths ms reynolds a singing and acting leading lady of an earlier era died at 84 in the throes of a mother grief.and for tens of thousands
of people who might have choked to death had they not been saved by his simple but ingenious maneuver the passing of henry j heimlich prompted not ju
st sympathy but even more gratitude'
```

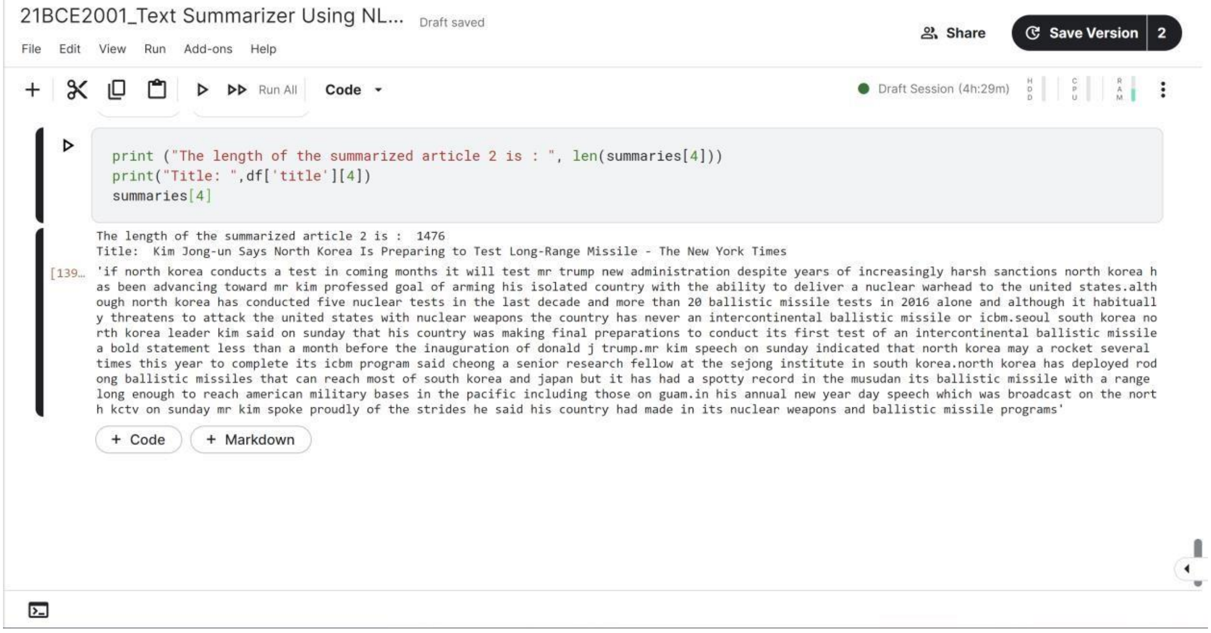
SAMPLE INPUT 2:



```python
print ("The length of the 2nd Test article is : ", len(df['article'][4]))
print("Title: ",df['title'][4])
df['article'][4]
```

```
The length of the 2nd Test article is :  4195
Title:  Kim Jong-un Says North Korea Is Preparing to Test Long-Range Missile - The New York Times
'SEOUL, South Korea  —   North Korea’s leader, Kim   said on Sunday that his country was making final preparations to conduct its first test of an in
tercontinental ballistic missile  —   a bold statement less than a month before the inauguration of   Donald J. Trump. Although North Korea has condu
cted five nuclear tests in the last decade and more than 20 ballistic missile tests in 2016 alone, and although it habitually threatens to attack the
United States with nuclear weapons, the country has never   an intercontinental ballistic missile, or ICBM. In his annual New Year’s Day speech, whic
h was broadcast on the North’s   KCTV on Sunday, Mr. Kim spoke proudly of the strides he said his country had made in its nuclear weapons and ballist
ic missile programs. He said North Korea would continue to bolster its weapons programs as long as the United States remained hostile and continued i
ts joint military exercises with South Korea. “We have reached the final stage in preparations to   an intercontinental ballistic rocket,” he said. A
nalysts in the region have said Mr. Kim might conduct another weapons test in coming months, taking advantage of leadership changes in the United Sta
tes and South Korea. Mr. Trump will be sworn in on Jan. 20. In South Korea, President Park   whose powers were suspended in a Parliamentary impeachme
nt on Dec. 9, is waiting for the Constitutional Court to rule on whether she should be formally removed from office or reinstated. If North Korea con
ducts a    test in coming months, it will test Mr. Trump’s new administration despite years of increasingly harsh sanctions, North Korea has been adv
ancing toward Mr. Kim’s professed goal of arming his isolated country with the ability to deliver a nuclear warhead to the United States. Mr. Kim’s s
peech on Sunday indicated that North Korea may   a   rocket several times this year to complete its ICBM program, said Cheong   a senior research fel
low at the Sejong Institute in South Korea. The first of such tests could come even before Mr. Trump’s inauguration, Mr. Cheong said. “We need to tak
e note of the fact that this is the first New Year’s speech where Kim   mentioned an intercontinental ballistic missile,” he said. In his speech, Mr.
Kim did not comment on Mr. Trump’s election. Doubt still runs deep that North Korea has mastered all the technology needed to build a reliable ICBM.
But analysts in the region said the North’s launchings of   rockets to put satellites into orbit in recent years showed that the country had cleared
some key technological hurdles. After the North’s satellite launch in February, South Korean defense officials said the Unha rocket used in the launc
h, if successfully reconfigured as a missile, could fly more than 7, 400 miles with a warhead of 1, 100 to 1, 300 pounds  —   far enough to reach mos
t of the United States. North Korea has deployed Rodong ballistic missiles that can reach most of South Korea and Japan, but it has had a spotty reco
rd in   the Musudan, its   ballistic missile with a range long enough to reach American military bases in the Pacific, including those on Guam. The N
orth has also claimed a series of successes in testing various ICBM technologies, although its claims cannot be verified and are often disputed by of
ficials and analysts in the region. It has said it could now make nuclear warheads small enough to fit onto a ballistic missile. It also claimed succ
```

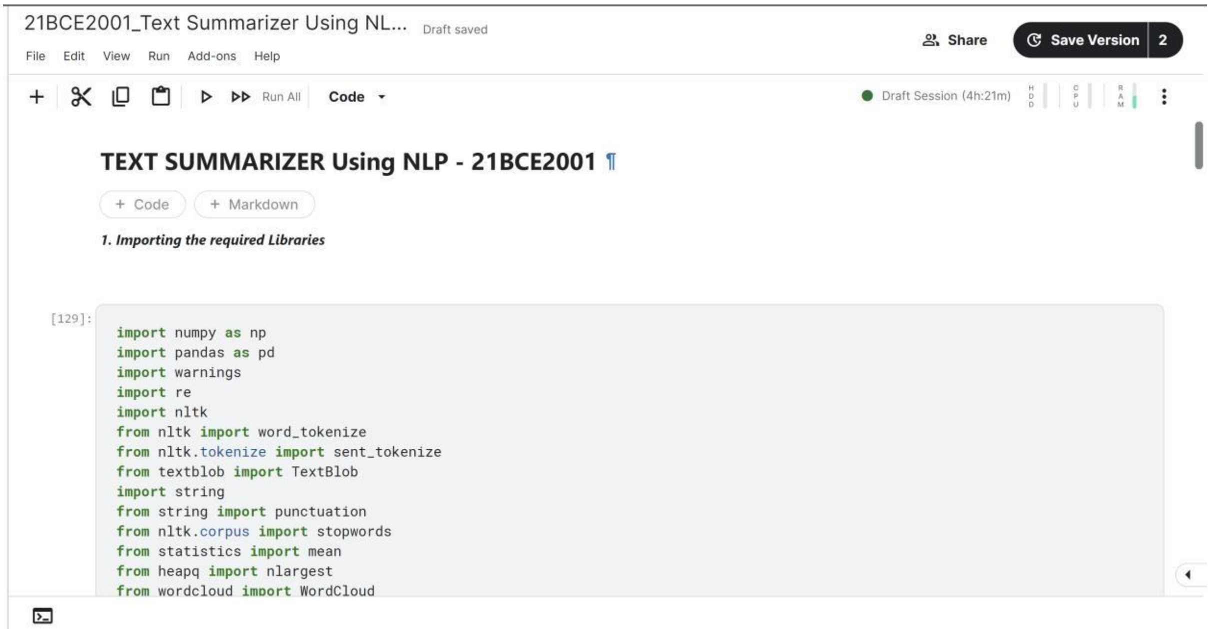
OUTPUT 2:



```python
print ("The length of the summarized article 2 is : ", len(summaries[4]))
print("Title: ",df['title'][4])
summaries[4]
```

```
The length of the summarized article 2 is :  1476
Title:  Kim Jong-un Says North Korea Is Preparing to Test Long-Range Missile - The New York Times
'if north korea conducts a test in coming months it will test mr trump new administration despite years of increasingly harsh sanctions north korea h
as been advancing toward mr kim professed goal of arming his isolated country with the ability to deliver a nuclear warhead to the united states.alth
ough north korea has conducted five nuclear tests in the last decade and more than 20 ballistic missile tests in 2016 alone and although it habituall
y threatens to attack the united states with nuclear weapons the country has never an intercontinental ballistic missile or icbm.seoul south korea no
rth korea leader kim said on sunday that his country was making final preparations to conduct its first test of an intercontinental ballistic missile
a bold statement less than a month before the inauguration of donald j trump.mr kim speech on sunday indicated that north korea may a rocket several
times this year to complete its icbm program said cheong a senior research fellow at the sejong institute in south korea.north korea has deployed rod
ong ballistic missiles that can reach most of south korea and japan but it has had a spotty record in the musudan its ballistic missile with a range
long enough to reach american military bases in the pacific including those on guam.in his annual new year day speech which was broadcast on the nort
h kctv on sunday mr kim spoke proudly of the strides he said his country had made in its nuclear weapons and ballistic missile programs'
```

OTHER SCREENSHOTS OF PROJECT::

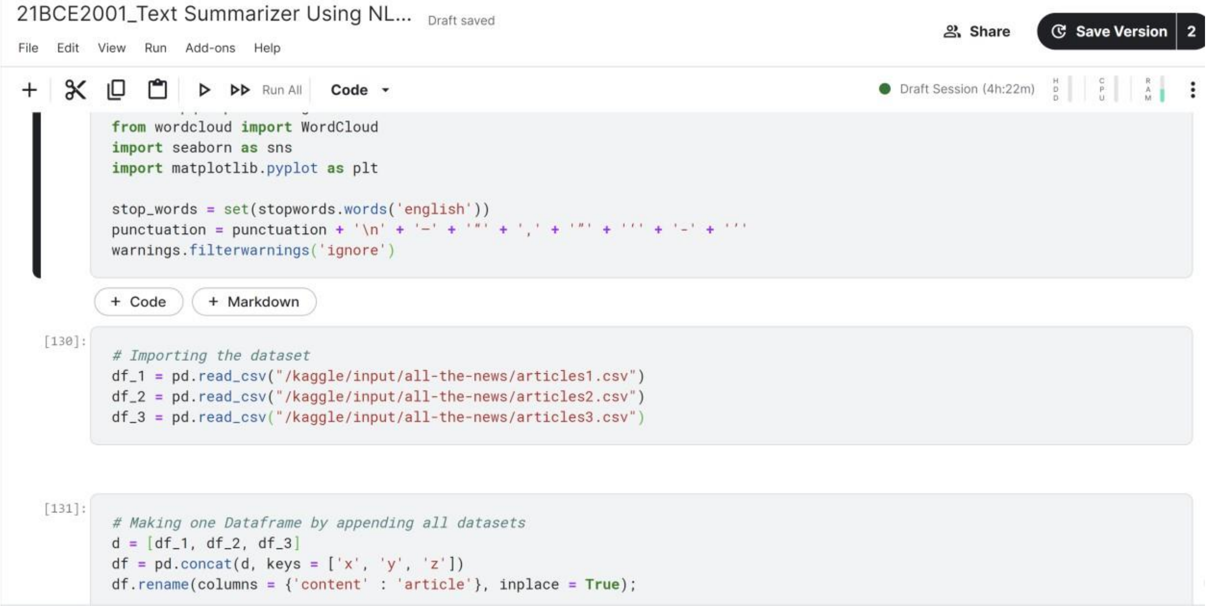


TEXT SUMMARIZER Using NLP - 21BCE2001 ¶

1. Importing the required Libraries

[129]:

```
import numpy as np
import pandas as pd
import warnings
import re
import nltk
from nltk import word_tokenize
from nltk.tokenize import sent_tokenize
from textblob import TextBlob
import string
from string import punctuation
from nltk.corpus import stopwords
from statistics import mean
from heapq import nlargest
from wordcloud import WordCloud
```



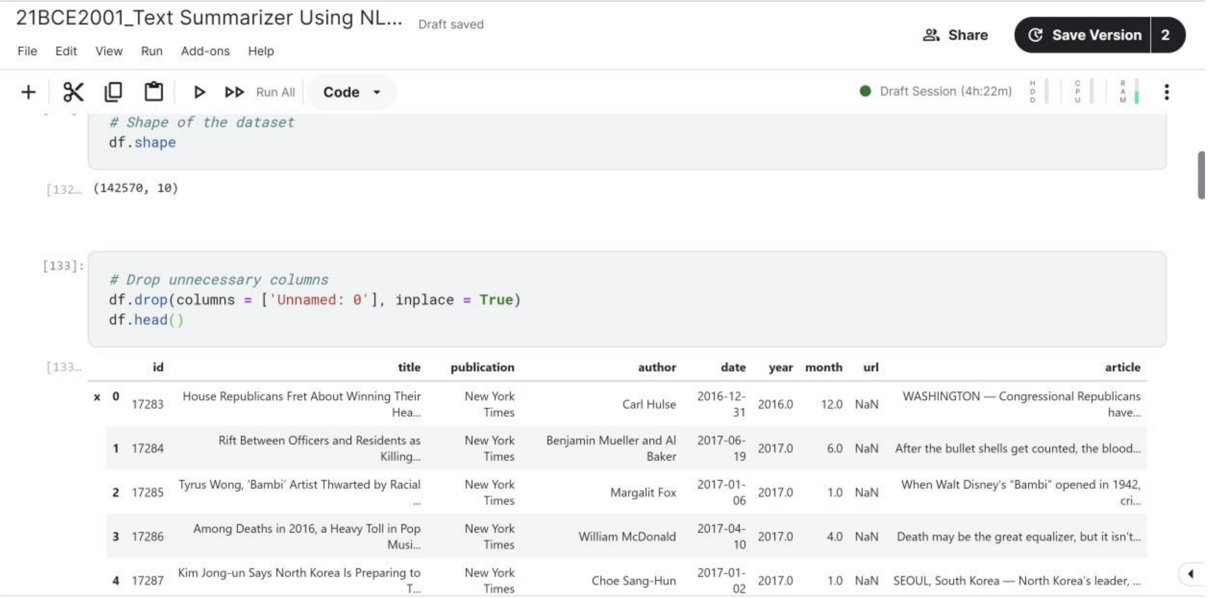
```
from wordcloud import WordCloud
import seaborn as sns
import matplotlib.pyplot as plt

stop_words = set(stopwords.words('english'))
punctuation = punctuation + '\n' + '—' + '“' + ',' + '”' + '‘' + '-' + '’'
warnings.filterwarnings('ignore')
```

[130]:

```
# Importing the dataset
df_1 = pd.read_csv("/kaggle/input/all-the-news/articles1.csv")
df_2 = pd.read_csv("/kaggle/input/all-the-news/articles2.csv")
df_3 = pd.read_csv("/kaggle/input/all-the-news/articles3.csv")
```

[131]:

```
# Making one Dataframe by appending all datasets
d = [df_1, df_2, df_3]
df = pd.concat(d, keys = ['x', 'y', 'z'])
df.rename(columns = {'content' : 'article'}, inplace = True);
```


21BCE2001_Text Summarizer Using NL...
Draft saved
Share
Save Version 2
File Edit View Run Add-ons Help
Run All
Code
Draft Session (4h:22m)

```
# Shape of the dataset
df.shape
```

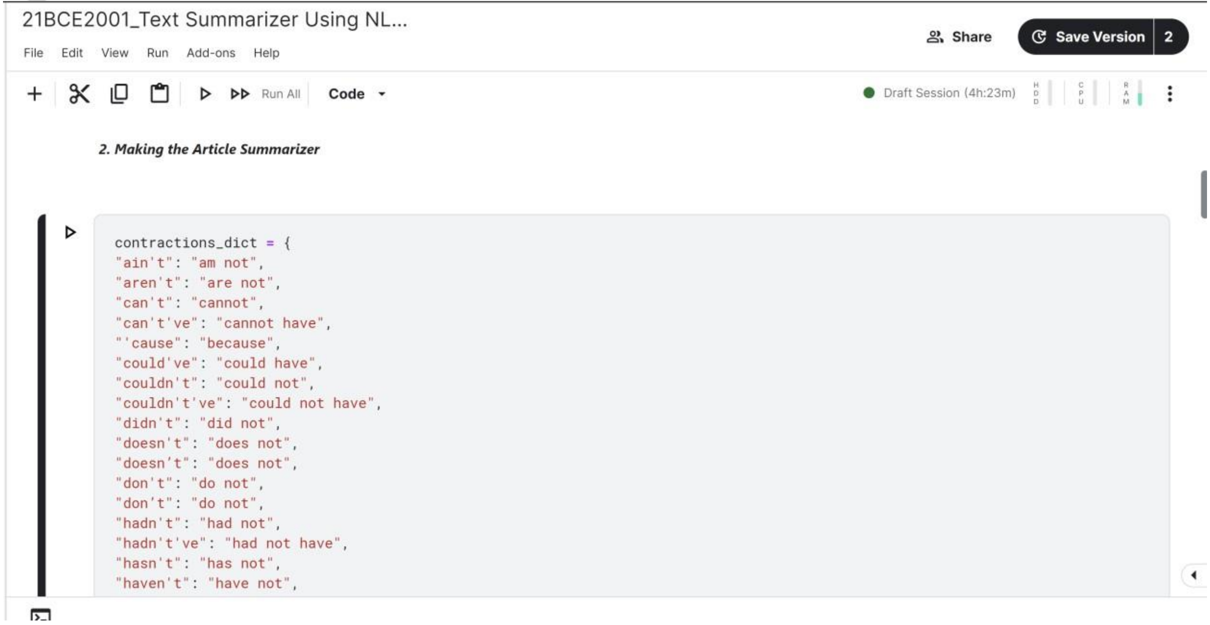
[132… (142570, 10)

[133]:

```
# Drop unnecessary columns
df.drop(columns = ['Unnamed: 0'], inplace = True)
df.head()
```

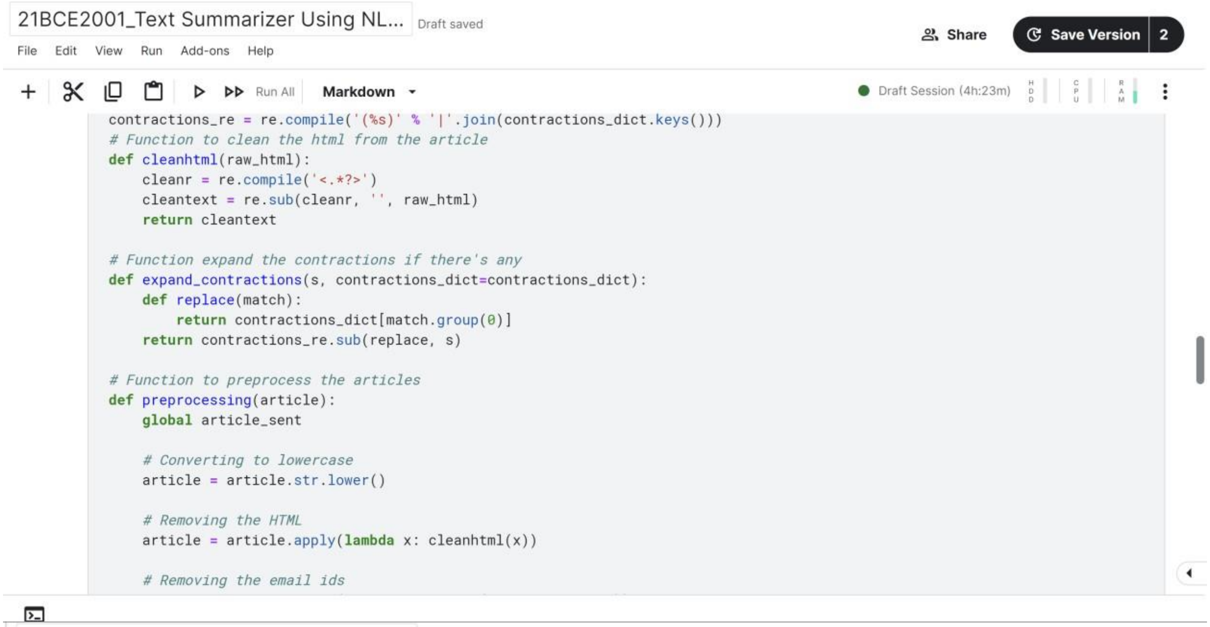
[133…

	id	title	publication	author	date	year	month	url	article
0	17283	House Republicans Fret About Winning Their Hea...	New York Times	Carl Hulse	2016-12-31	2016.0	12.0	NaN	WASHINGTON — Congressional Republicans have...
1	17284	Rift Between Officers and Residents as Killing...	New York Times	Benjamin Mueller and Al Baker	2017-06-19	2017.0	6.0	NaN	After the bullet shells get counted, the blood...
2	17285	Tyrus Wong, 'Bambi' Artist Thwarted by Racial ...	New York Times	Margalit Fox	2017-01-06	2017.0	1.0	NaN	When Walt Disney's "Bambi" opened in 1942, cri...
3	17286	Among Deaths in 2016, a Heavy Toll in Pop Musi...	New York Times	William McDonald	2017-04-10	2017.0	4.0	NaN	Death may be the great equalizer, but it isn't...
4	17287	Kim Jong-un Says North Korea Is Preparing to T...	New York Times	Choe Sang-Hun	2017-01-02	2017.0	1.0	NaN	SEOUL, South Korea — North Korea's leader, ...

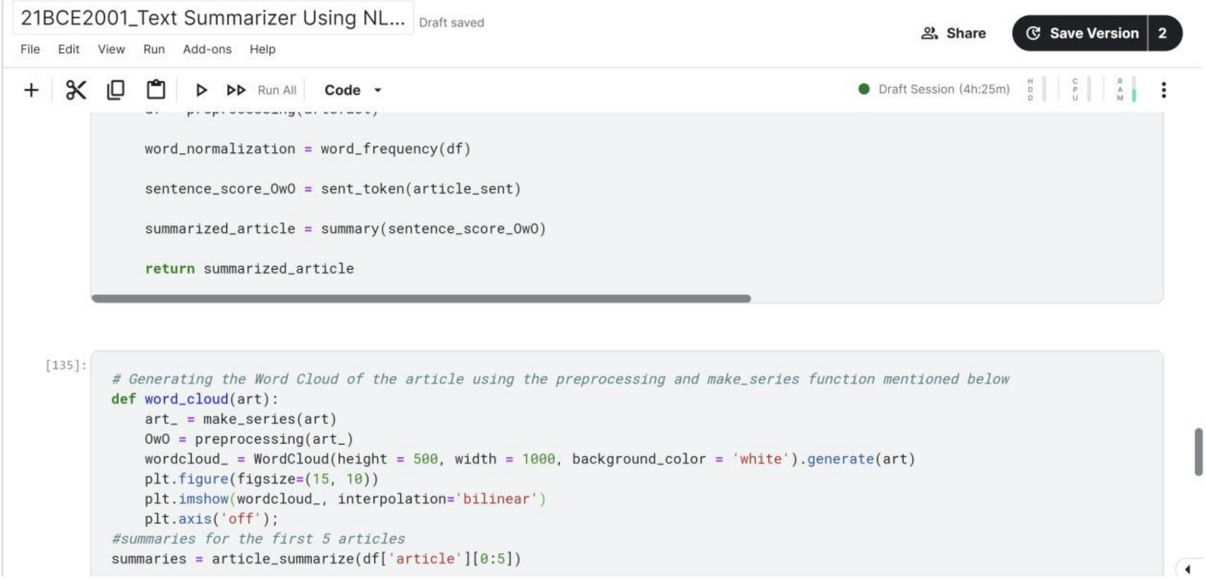

21BCE2001_Text Summarizer Using NL...
Share
Save Version 2
File Edit View Run Add-ons Help
Run All
Code
Draft Session (4h:23m)

2. Making the Article Summarizer

```
contractions_dict = {
"ain't": "am not",
"aren't": "are not",
"can't": "cannot",
"can't've": "cannot have",
"'cause": "because",
"could've": "could have",
"couldn't": "could not",
"couldn't've": "could not have",
"didn't": "did not",
"doesn't": "does not",
"doesn't": "does not",
"don't": "do not",
"don't": "do not",
"hadn't": "had not",
"hadn't've": "had not have",
"hasn't": "has not",
"haven't": "have not",
```



```
contractions_re = re.compile('(%s)' % '|'.join(contractions_dict.keys()))
# Function to clean the html from the article
def cleanhtml(raw_html):
    cleanr = re.compile('<.*?>')
    cleantext = re.sub(cleanr, '', raw_html)
    return cleantext

# Function expand the contractions if there's any
def expand_contractions(s, contractions_dict=contractions_dict):
    def replace(match):
        return contractions_dict[match.group(0)]
    return contractions_re.sub(replace, s)

# Function to preprocess the articles
def preprocessing(article):
    global article_sent

    # Converting to lowercase
    article = article.str.lower()

    # Removing the HTML
    article = article.apply(lambda x: cleanhtml(x))

    # Removing the email ids
```



```
    df = preprocessing(article)

    word_normalization = word_frequency(df)

    sentence_score_OwO = sent_token(article_sent)

    summarized_article = summary(sentence_score_OwO)

    return summarized_article
```

[135]:

```
# Generating the Word Cloud of the article using the preprocessing and make_series function mentioned below
def word_cloud(art):
    art_ = make_series(art)
    OwO = preprocessing(art_)
    wordcloud_ = WordCloud(height = 500, width = 1000, background_color = 'white').generate(art)
    plt.figure(figsize=(15, 10))
    plt.imshow(wordcloud_, interpolation='bilinear')
    plt.axis('off');
#summaries for the first 5 articles
summaries = article_summarize(df['article'][0:5])
```