Fake News Detection using NLP

PHASE V PROJECT SUBMISSION

AbijithKrishna

720921104003

FAKE NEWS DETECTION using (NLP):

Fake news detection using Natural Language Processing (NLP)

Here's an overview of the steps involved in fake news detection using NLP:

1. Data Collection:

   • Gather a large dataset of news articles or text data, including both fake and real news articles. Reliable sources for labeled datasets include websites like Snopes, PolitiFact, and FactCheck.org.

2. Text Preprocessing:

   • Clean and preprocess the text data to remove noise and irrelevant information. Common preprocessing steps include: • Tokenization: Splitting text into words or tokens. • Lowercasing: Converting all text to lowercase. • Removing stopwords: Eliminating common words that don't carry much information. • Stemming or Lemmatization: Reducing words to their base forms. • Removing special characters and punctuation.

3. Feature Extraction:

   • Transform the preprocessed text data into numerical features that can be used for machine learning models. Common techniques include: • TF-IDF (Term Frequency-Inverse Document Frequency): This method assigns a weight to each term based on its frequency in a document relative to its frequency in the entire corpus. • Word Embeddings: Use pre-trained word embeddings like Word2Vec, GloVe, or FastText to represent words as dense vectors. • Document Embeddings: Aggregate word embeddings to represent entire documents.

4. Model Selection:

   • Choose a machine learning or deep learning model for fake news detection. Common choices include: • Logistic Regression • Naive Bayes • Support Vector Machines • Recurrent Neural Networks (RNNs) • Convolutional Neural Networks (CNNs) • Transformer-based models like BERT or GPT-3.

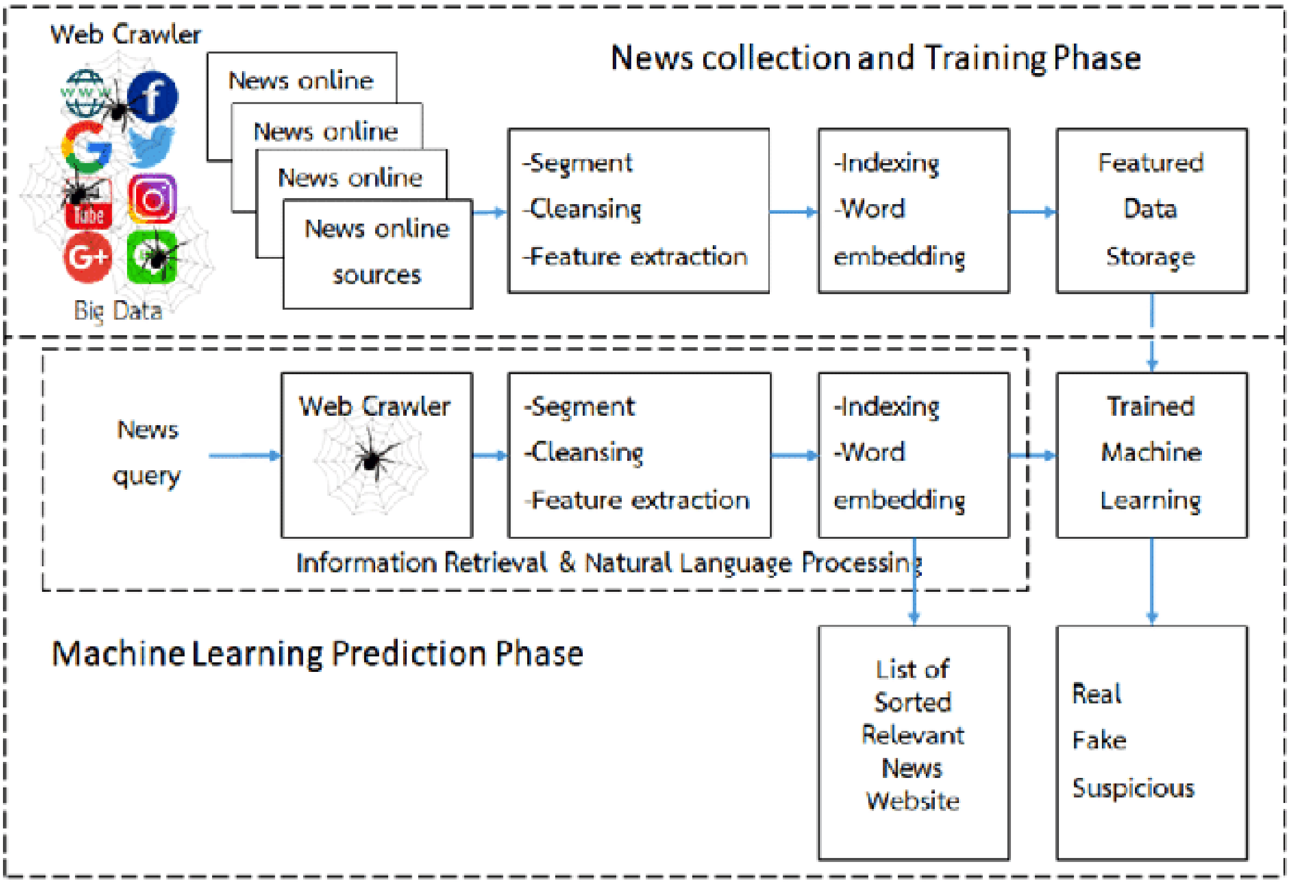
5. Model Training:

   • Split your dataset into training and testing sets. • Train the selected model on the training data using appropriate evaluation metrics (e.g., accuracy, F1-score). • Fine-tune hyperparameters to optimize performance.

6. Model Evaluation:

- Evaluate the model on the testing dataset using relevant evaluation metrics.
- Consider other metrics like precision, recall, and ROC-AUC, as accuracy alone may not be sufficient for imbalanced datasets.

7. Post-processing:
   - Apply post-processing techniques to refine the model's predictions. For example, you can set a threshold for classifying articles as fake or real based on the model's confidence score.
8. Deployment:
   - Once you have a well-performing model, you can deploy it as part of a fake news detection system. This system can be integrated into social media platforms or news aggregators to automatically flag potentially fake news articles.
9. Continuous Improvement:
   - Continuously update and retrain your model as new data becomes available to adapt to evolving patterns of misinformation. Keep in mind that fake news detection is a challenging problem, and no single model or approach is foolproof. It's essential to combine NLP techniques with human expertise and critical thinking to combat the spread of misinformation effectively.



DETECTION MISINFORMATION AND FAKE NEWA THROUGH THE POWER OF NLP:

Here are some inventive strategies in the realm of fake news detection using NLP:

1. Linguistic DNA Profiling:

   - Delve deep into linguistic DNA, dissecting elements like sentence structure, grammatical anomalies, and vocabulary quirks. Each fake news piece leaves its unique linguistic imprint.

2. Contextual Semantic Analysis:

   - Move beyond mere keyword analysis. Employ NLP to discern the subtle nuances of context, semantics, and tone. Fake news often skews meaning or distorts the facts subtly.

3. Multimodal Fusion:

   - Combine textual analysis with image and video analysis. Many fake news stories circulate through multimedia content, and NLP can be used to cross-verify text against accompanying media.

4. Cross-Lingual Verification:

   - Expand horizons by assessing content in multiple languages. Fake news knows no linguistic boundaries, and NLP can help bridge the gap by cross-referencing news in different languages.

5. Real-Time Verification Ecosystem:

   - Build an ecosystem of real-time fact-checking and source verification. NLP can be integrated into this system to continuously monitor and validate news stories.

6. Neural Summarization and Comparison:

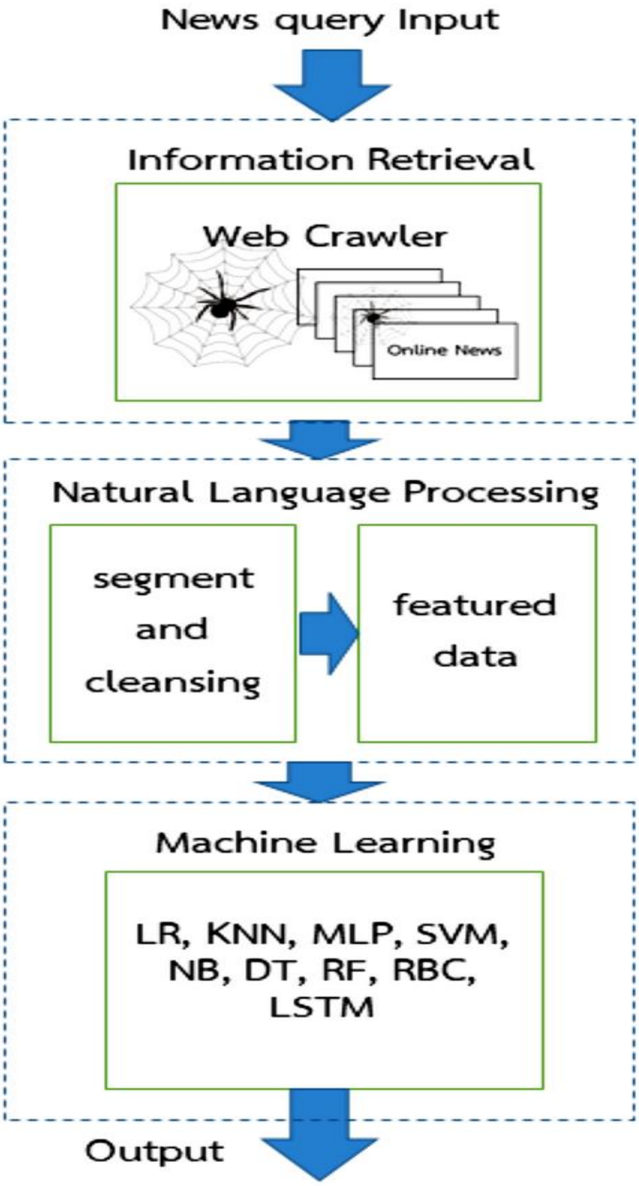
   - Use advanced summarization techniques driven by neural networks to condense news articles and then compare them with known factual sources. Fake news often lacks comprehensive details.

7. Disinformation Network Mapping:

- Go beyond individual articles and map the web of disinformation networks. NLP can identify patterns in how fake news spreads and connects.

8. Behavioral and Sentiment Analysis:
   - Examine not only the content but also the behavioral patterns of those spreading the news. NLP can detect sentiment shifts and suspicious sharing practices.
9. Explainable AI for Trustworthiness Scores:
   - Develop AI models that provide transparent scores indicating the trustworthiness of a news piece. Explainable AI helps users understand why a piece of news is flagged as suspicious.
10. Quantum Cryptography for Metadata Verification:

    Leverage cutting-edge technology like quantum cryptography to ensure the integrity of metadata. This can help identify tampered publication dates and locations.



STRATEGIES OUTLINED IN YOUR TEXT ARE INDEED INNOVATIVE APPROACHE IN THE REAL OF FAKE NEWS DETECTION Using NLP:

Here's a closer look at each of these strategies:

Linguistic DNA Profiling: Analyzing the linguistic patterns and anomalies in text is a foundational NLP technique. Each fake news story can indeed leave a unique linguistic imprint, which can be detected using NLP.

Contextual Semantic Analysis: This goes beyond simple keyword matching by understanding the context and semantics of the text. Fake news often manipulates the meaning of words and phrases, and NLP can help detect these subtle distortions.

Multimodal Fusion: Combining text analysis with image and video analysis is crucial, as fake news often circulates through various media. Using NLP for cross-verifying text against multimedia content is a valuable approach.

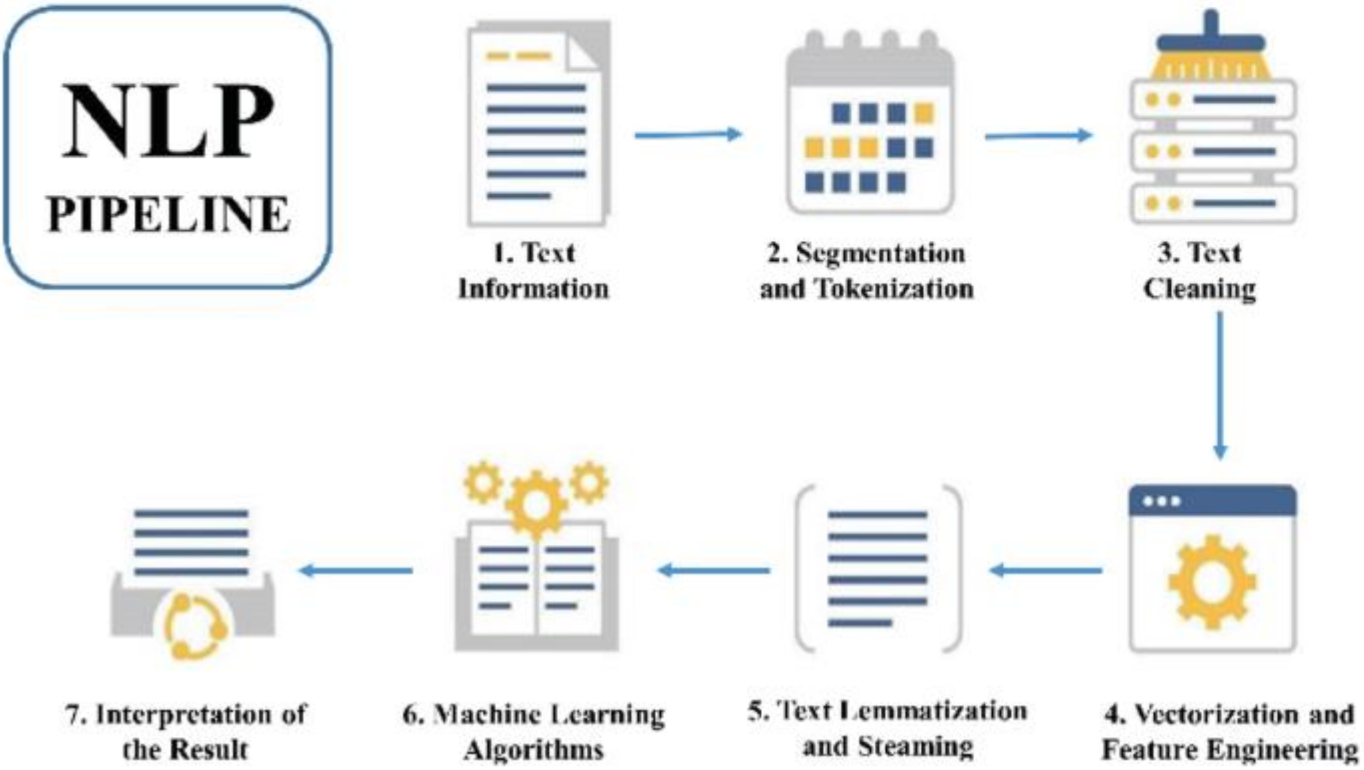
Cross-Lingual Verification: Misinformation is not limited by language, so it's important to assess content in multiple languages. NLP can be used to translate and cross-reference news in different languages.

Real-Time Verification Ecosystem: Continuously monitoring and verifying news stories in real-time is an effective way to combat fake news. NLP can play a role in automating this process.

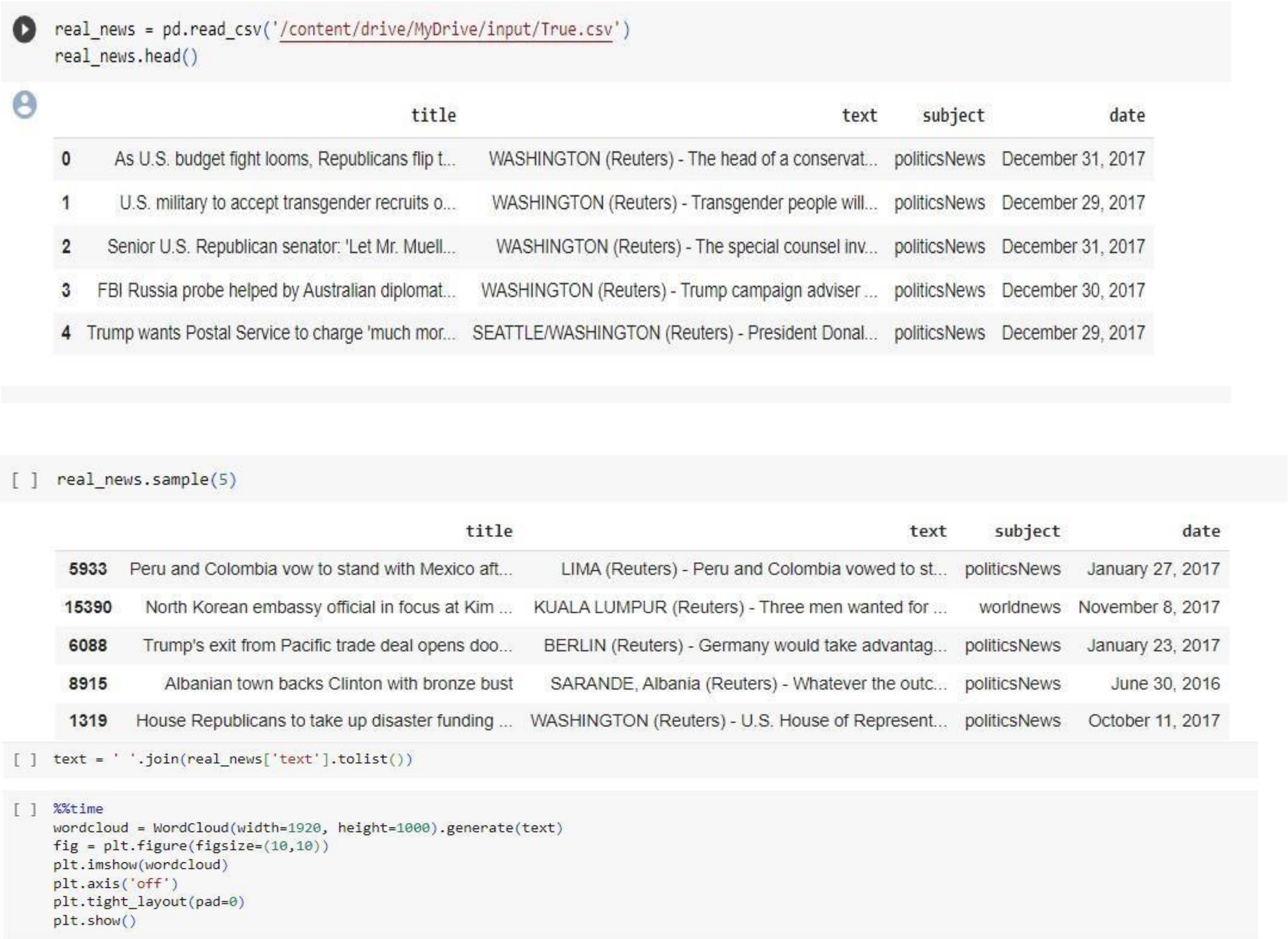
Neural Summarization and Comparison: Summarization techniques driven by neural networks can help condense news articles and make it easier to compare them with known factual sources. Disinformation Network Mapping: NLP can help identify the patterns in how fake news spreads and connects.

This can be valuable for understanding the dissemination of false information. Behavioral and Sentiment Analysis: Analyzing the behavior and sentiment of those spreading fake news can provide additional clues for detection. NLP can help in tracking sentiment shifts and suspicious sharing practices.

Explainable AI for Trustworthiness Scores: Providing transparent trustworthiness scores is essential for building user trust in fake news detection systems. Explainable AI can help users understand why a piece of news is flagged as suspicious.


NLP
PIPELINE
1. Text Information
2. Segmentation and Tokenization
3. Text Cleaning
4. Vectorization and Feature Engineering
5. Text Lemmatization and Steaming
6. Machine Learning Algorithms
7. Interpretation of the Result

Word cloud for True News:



```
real_news = pd.read_csv('/content/drive/MyDrive/input/True.csv')
real_news.head()
```

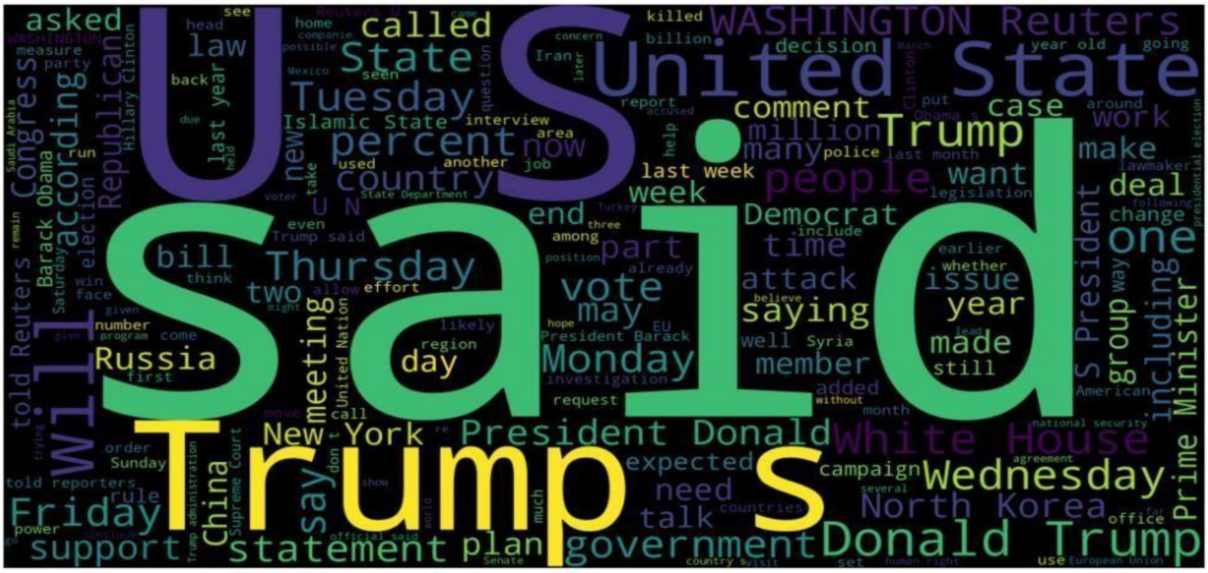
	title	text	subject	date
0	As U.S. budget fight looms, Republicans flip t...	WASHINGTON (Reuters) - The head of a conservat...	politicsNews	December 31, 2017
1	U.S. military to accept transgender recruits o...	WASHINGTON (Reuters) - Transgender people will...	politicsNews	December 29, 2017
2	Senior U.S. Republican senator: 'Let Mr. Muell...	WASHINGTON (Reuters) - The special counsel inv...	politicsNews	December 31, 2017
3	FBI Russia probe helped by Australian diplomat...	WASHINGTON (Reuters) - Trump campaign adviser ...	politicsNews	December 30, 2017
4	Trump wants Postal Service to charge 'much mor...	SEATTLE/WASHINGTON (Reuters) - President Donal...	politicsNews	December 29, 2017

```
[ ] real_news.sample(5)
```

	title	text	subject	date
5933	Peru and Colombia vow to stand with Mexico aft...	LIMA (Reuters) - Peru and Colombia vowed to st...	politicsNews	January 27, 2017
15390	North Korean embassy official in focus at Kim ...	KUALA LUMPUR (Reuters) - Three men wanted for ...	worldnews	November 8, 2017
6088	Trump's exit from Pacific trade deal opens doo...	BERLIN (Reuters) - Germany would take advantag...	politicsNews	January 23, 2017
8915	Albanian town backs Clinton with bronze bust	SARANDE, Albania (Reuters) - Whatever the outc...	politicsNews	June 30, 2016
1319	House Republicans to take up disaster funding ...	WASHINGTON (Reuters) - U.S. House of Represent...	politicsNews	October 11, 2017

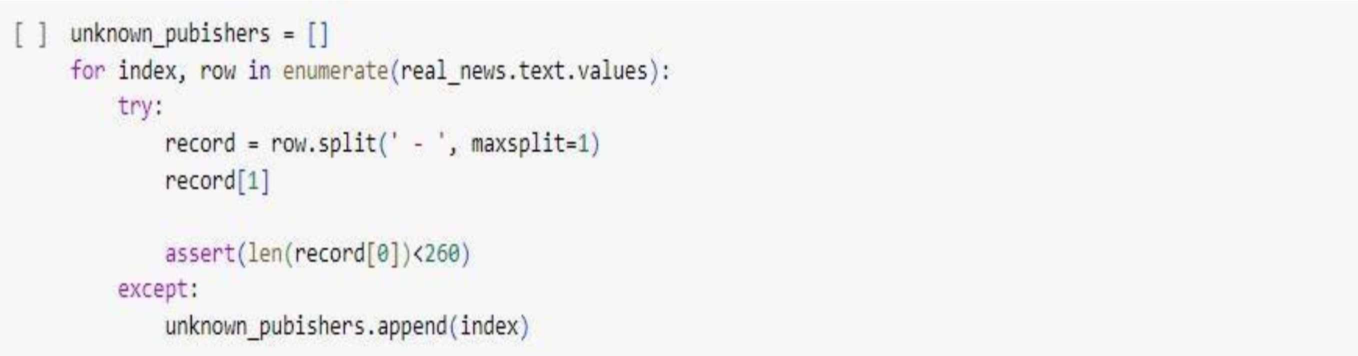
```
[ ] text = ' '.join(real_news['text'].tolist())
```

```
[ ] %%time
    wordcloud = WordCloud(width=1920, height=1000).generate(text)
    fig = plt.figure(figsize=(10,10))
    plt.imshow(wordcloud)
    plt.axis('off')
    plt.tight_layout(pad=0)
    plt.show()
```

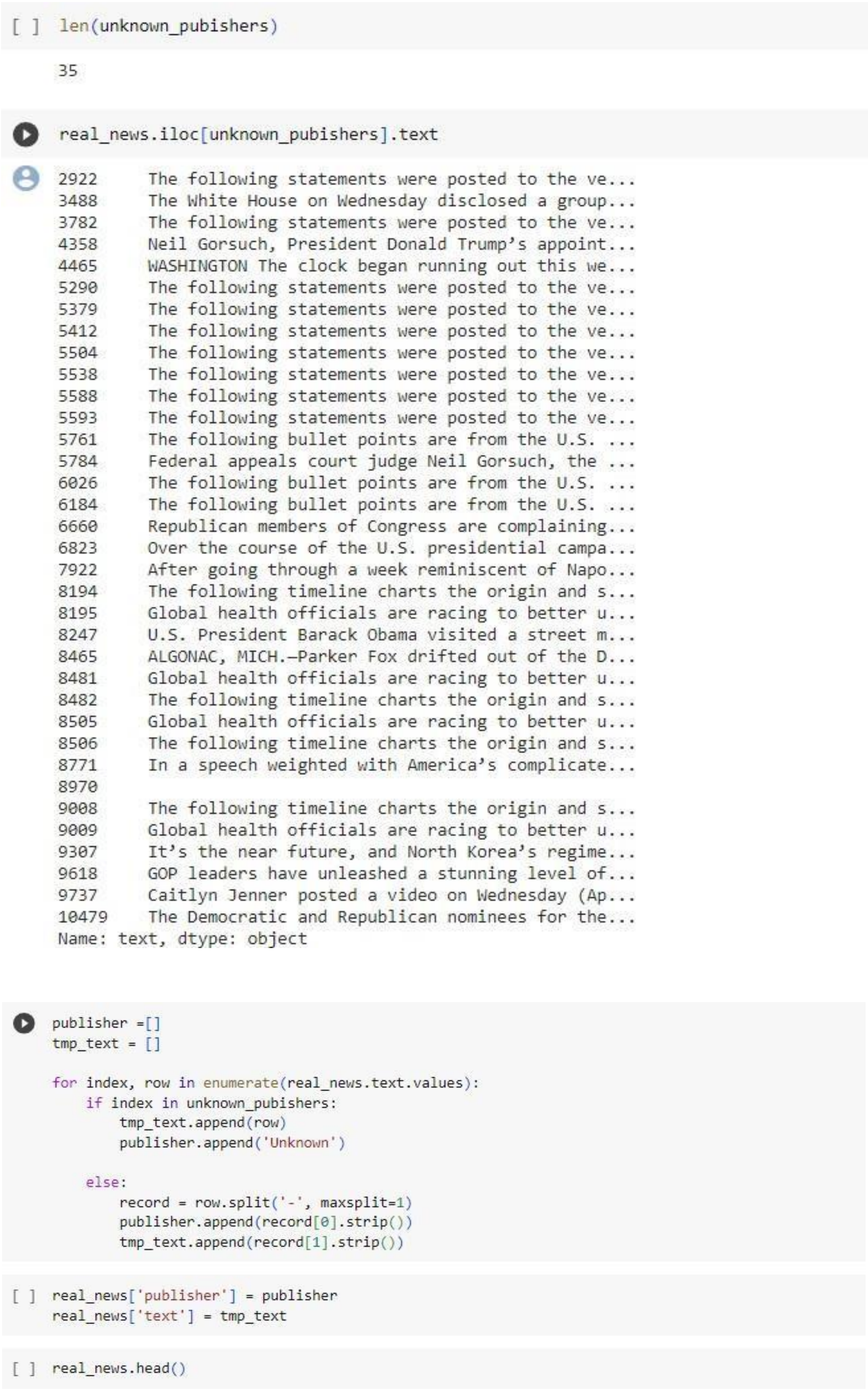


```
CPU times: user 38.5 s, sys: 2.83 s, total: 41.3 s
Wall time: 41.4 s
```

Let’s create a list of news lists in real_news.csv with unknown publishers by using the following code snippets



```
unknown_pubishers = []
for index, row in enumerate(real_news.text.values):
    try:
        record = row.split(' - ', maxsplit=1)
        record[1]

        assert(len(record[0])<260)
    except:
        unknown_pubishers.append(index)
```



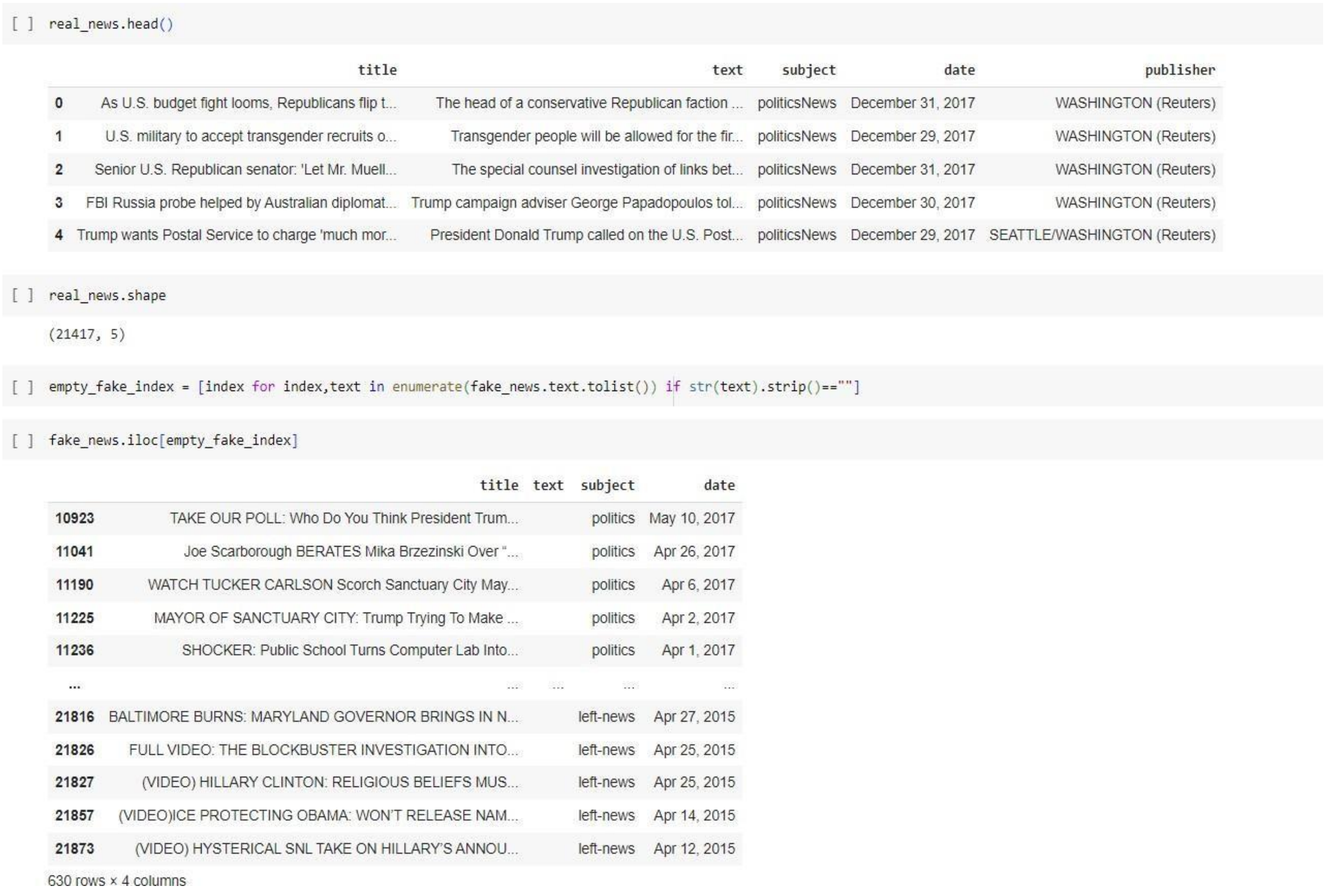
```
[ ] len(unknown_pubishers)

    35
```

```
real_news.iloc[unknown_pubishers].text

2922     The following statements were posted to the ve...
3488     The White House on Wednesday disclosed a group...
3782     The following statements were posted to the ve...
4358     Neil Gorsuch, President Donald Trump’s appoint...
4465     WASHINGTON The clock began running out this we...
5290     The following statements were posted to the ve...
5379     The following statements were posted to the ve...
5412     The following statements were posted to the ve...
5504     The following statements were posted to the ve...
5538     The following statements were posted to the ve...
5588     The following statements were posted to the ve...
5593     The following statements were posted to the ve...
5761     The following bullet points are from the U.S. ...
5784     Federal appeals court judge Neil Gorsuch, the ...
6026     The following bullet points are from the U.S. ...
6184     The following bullet points are from the U.S. ...
6660     Republican members of Congress are complaining...
6823     Over the course of the U.S. presidential campa...
7922     After going through a week reminiscent of Napo...
8194     The following timeline charts the origin and s...
8195     Global health officials are racing to better u...
8247     U.S. President Barack Obama visited a street m...
8465     ALGONAC, MICH.—Parker Fox drifted out of the D...
8481     Global health officials are racing to better u...
8482     The following timeline charts the origin and s...
8505     Global health officials are racing to better u...
8506     The following timeline charts the origin and s...
8771     In a speech weighted with America’s complicate...
8970
9008     The following timeline charts the origin and s...
9009     Global health officials are racing to better u...
9307     It’s the near future, and North Korea’s regime...
9618     GOP leaders have unleashed a stunning level of...
9737     Caitlyn Jenner posted a video on Wednesday (Ap...
10479    The Democratic and Republican nominees for the...
Name: text, dtype: object
```

```
publisher =[]
tmp_text = []

for index, row in enumerate(real_news.text.values):
    if index in unknown_pubishers:
        tmp_text.append(row)
        publisher.append('Unknown')

    else:
        record = row.split('-', maxsplit=1)
        publisher.append(record[0].strip())
        tmp_text.append(record[1].strip())
```

```
[ ] real_news['publisher'] = publisher
    real_news['text'] = tmp_text
```

```
[ ] real_news.head()
```



```
[ ] real_news.head()
```

	title	text	subject	date	publisher
0	As U.S. budget fight looms, Republicans flip t...	The head of a conservative Republican faction ...	politicsNews	December 31, 2017	WASHINGTON (Reuters)
1	U.S. military to accept transgender recruits o...	Transgender people will be allowed for the fir...	politicsNews	December 29, 2017	WASHINGTON (Reuters)
2	Senior U.S. Republican senator: 'Let Mr. Muell...	The special counsel investigation of links bet...	politicsNews	December 31, 2017	WASHINGTON (Reuters)
3	FBI Russia probe helped by Australian diplomat...	Trump campaign adviser George Papadopoulos tol...	politicsNews	December 30, 2017	WASHINGTON (Reuters)
4	Trump wants Postal Service to charge 'much mor...	President Donald Trump called on the U.S. Post...	politicsNews	December 29, 2017	SEATTLE/WASHINGTON (Reuters)

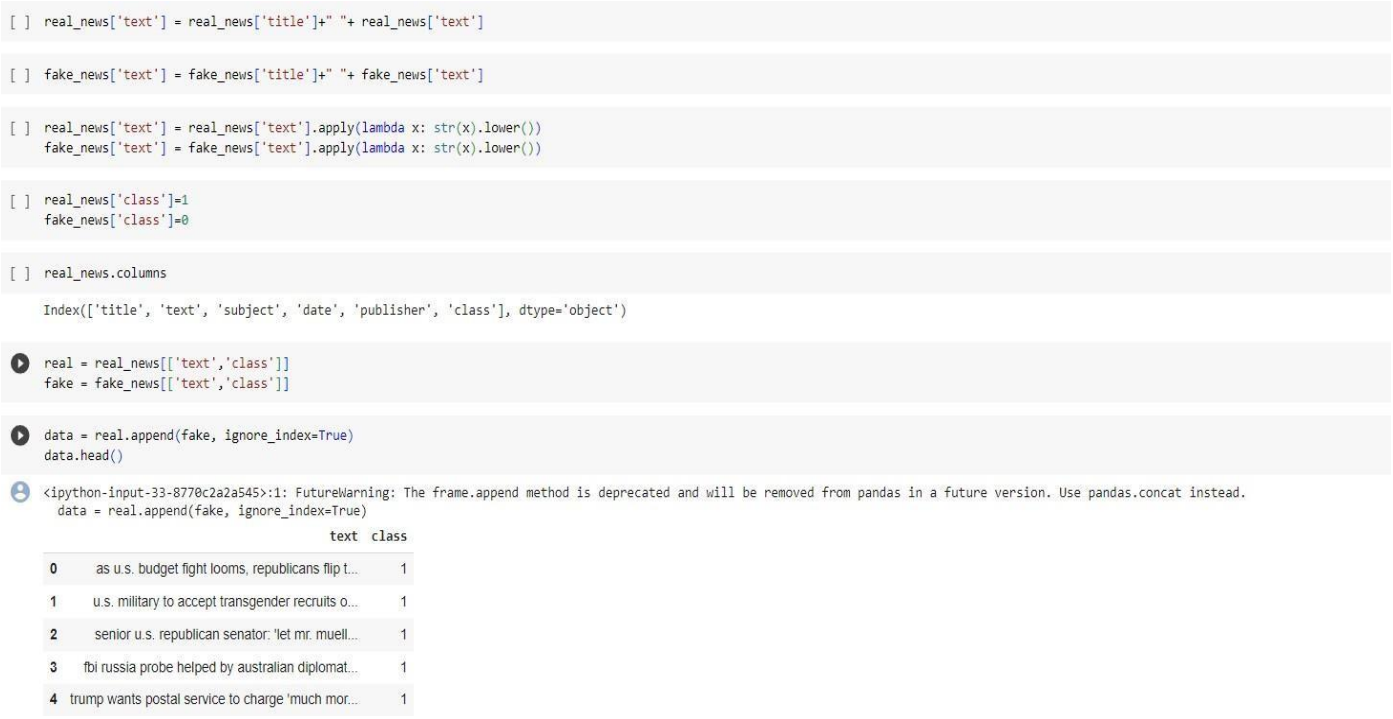
```
[ ] real_news.shape

    (21417, 5)
```

```
[ ] empty_fake_index = [index for index,text in enumerate(fake_news.text.tolist()) if str(text).strip()==""]
```

```
[ ] fake_news.iloc[empty_fake_index]
```

	title	text	subject	date
10923	TAKE OUR POLL: Who Do You Think President Trum...		politics	May 10, 2017
11041	Joe Scarborough BERATES Mika Brzezinski Over “...		politics	Apr 26, 2017
11190	WATCH TUCKER CARLSON Scorch Sanctuary City May...		politics	Apr 6, 2017
11225	MAYOR OF SANCTUARY CITY: Trump Trying To Make ...		politics	Apr 2, 2017
11236	SHOCKER: Public School Turns Computer Lab Into...		politics	Apr 1, 2017
...	...	...	...	...
21816	BALTIMORE BURNS: MARYLAND GOVERNOR BRINGS IN N...		left-news	Apr 27, 2015
21826	FULL VIDEO: THE BLOCKBUSTER INVESTIGATION INTO...		left-news	Apr 25, 2015
21827	(VIDEO) HILLARY CLINTON: RELIGIOUS BELIEFS MUS...		left-news	Apr 25, 2015
21857	(VIDEO)ICE PROTECTING OBAMA: WON'T RELEASE NAM...		left-news	Apr 14, 2015
21873	(VIDEO) HYSTERICAL SNL TAKE ON HILLARY'S ANNOU...		left-news	Apr 12, 2015

630 rows × 4 columns



```
[ ] real_news['text'] = real_news['title']+" "+ real_news['text']
```

```
[ ] fake_news['text'] = fake_news['title']+" "+ fake_news['text']
```

```
[ ] real_news['text'] = real_news['text'].apply(lambda x: str(x).lower())
    fake_news['text'] = fake_news['text'].apply(lambda x: str(x).lower())
```

```
[ ] real_news['class']=1
    fake_news['class']=0
```

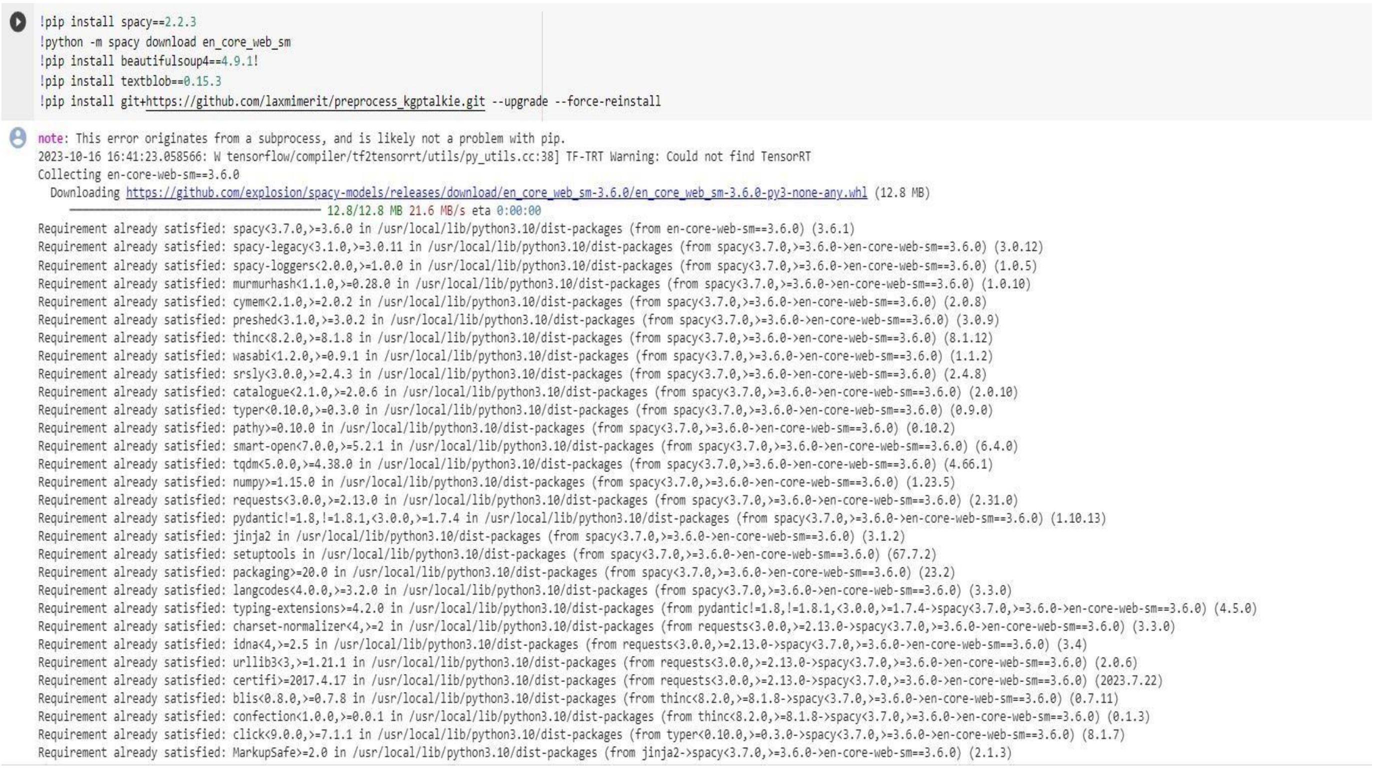
```
[ ] real_news.columns

    Index(['title', 'text', 'subject', 'date', 'publisher', 'class'], dtype='object')
```

```
real = real_news[['text','class']]
fake = fake_news[['text','class']]
```

```
data = real.append(fake, ignore_index=True)
data.head()
```

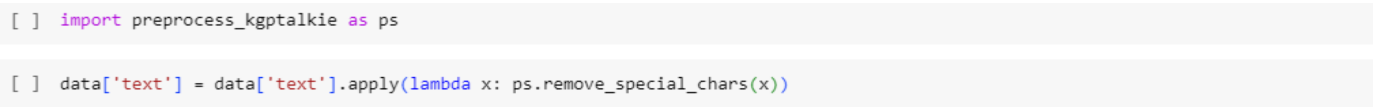
```
<ipython-input-33-8770c2a2a545>:1: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.
  data = real.append(fake, ignore_index=True)
```

	text	class
0	as u.s. budget fight looms, republicans flip t...	1
1	u.s. military to accept transgender recruits o...	1
2	senior u.s. republican senator: 'let mr. muell...	1
3	fbi russia probe helped by australian diplomat...	1
4	trump wants postal service to charge 'much mor...	1



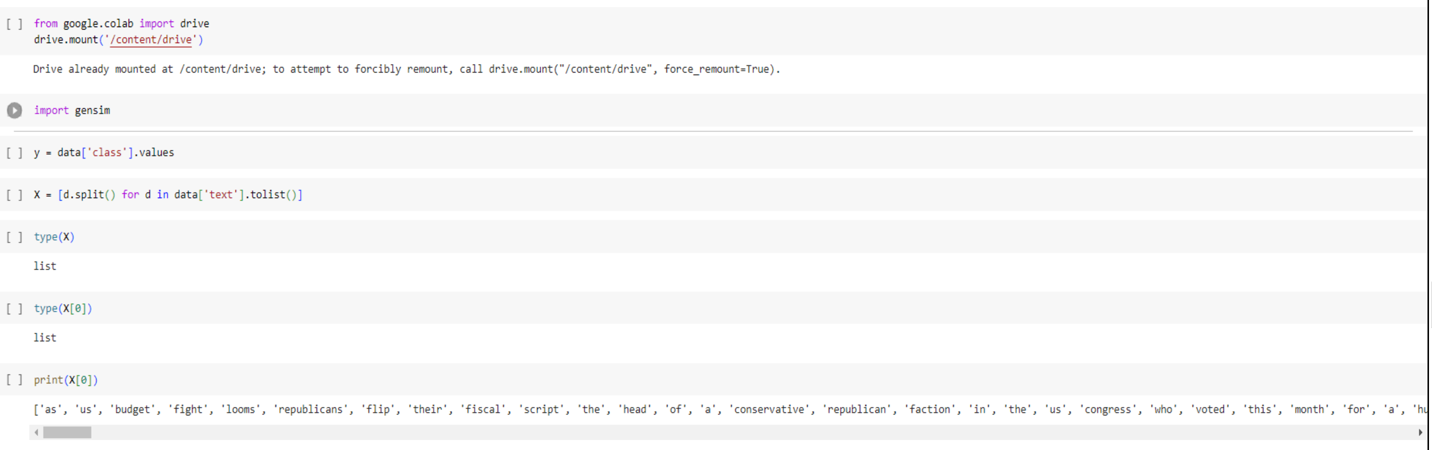
```
!pip install spacy==2.2.3
!python -m spacy download en_core_web_sm
!pip install beautifulsoup4==4.9.1!
!pip install textblob==0.15.3
!pip install git+https://github.com/laxmimerit/preprocess_kgptalkie.git --upgrade --force-reinstall
```

```
note: This error originates from a subprocess, and is likely not a problem with pip.
2023-10-16 16:41:23.058566: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
Collecting en-core-web-sm==3.6.0
  Downloading https://github.com/explosion/spacy-models/releases/download/en_core_web_sm-3.6.0/en_core_web_sm-3.6.0-py3-none-any.whl (12.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.8/12.8 MB 21.6 MB/s eta 0:00:00
Requirement already satisfied: spacy<3.7.0,>=3.6.0 in /usr/local/lib/python3.10/dist-packages (from en-core-web-sm==3.6.0) (3.6.1)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm==3.6.0) (3.0.12)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm==3.6.0) (1.0.5)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm==3.6.0) (1.0.10)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm==3.6.0) (2.0.8)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm==3.6.0) (3.0.9)
Requirement already satisfied: thinc<8.2.0,>=8.1.8 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm==3.6.0) (8.1.12)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm==3.6.0) (1.1.2)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm==3.6.0) (2.4.8)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm==3.6.0) (2.0.10)
Requirement already satisfied: typer<0.10.0,>=0.3.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm==3.6.0) (0.9.0)
Requirement already satisfied: pathy>=0.10.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm==3.6.0) (0.10.2)
Requirement already satisfied: smart-open<7.0.0,>=5.2.1 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm==3.6.0) (6.4.0)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm==3.6.0) (4.66.1)
Requirement already satisfied: numpy>=1.15.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm==3.6.0) (1.23.5)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm==3.6.0) (2.31.0)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm==3.6.0) (1.10.13)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm==3.6.0) (3.1.2)
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm==3.6.0) (67.7.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm==3.6.0) (23.2)
Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->en-core-web-sm==3.6.0) (3.3.0)
Requirement already satisfied: typing-extensions>=4.2.0 in /usr/local/lib/python3.10/dist-packages (from pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy<3.7.0,>=3.6.0->en-core-web-sm==3.6.0) (4.5.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.0,>=2.13.0->spacy<3.7.0,>=3.6.0->en-core-web-sm==3.6.0) (3.3.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.0,>=2.13.0->spacy<3.7.0,>=3.6.0->en-core-web-sm==3.6.0) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.0,>=2.13.0->spacy<3.7.0,>=3.6.0->en-core-web-sm==3.6.0) (2.0.6)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.0,>=2.13.0->spacy<3.7.0,>=3.6.0->en-core-web-sm==3.6.0) (2023.7.22)
Requirement already satisfied: blis<0.8.0,>=0.7.8 in /usr/local/lib/python3.10/dist-packages (from thinc<8.2.0,>=8.1.8->spacy<3.7.0,>=3.6.0->en-core-web-sm==3.6.0) (0.7.11)
Requirement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.10/dist-packages (from thinc<8.2.0,>=8.1.8->spacy<3.7.0,>=3.6.0->en-core-web-sm==3.6.0) (0.1.3)
Requirement already satisfied: click<9.0.0,>=7.1.1 in /usr/local/lib/python3.10/dist-packages (from typer<0.10.0,>=0.3.0->spacy<3.7.0,>=3.6.0->en-core-web-sm==3.6.0) (8.1.7)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->spacy<3.7.0,>=3.6.0->en-core-web-sm==3.6.0) (2.1.3)
```



```
import preprocess_kgptalkie as ps
```

```
data['text'] = data['text'].apply(lambda x: ps.remove_special_chars(x))
```



```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
import gensim
```

```
y = data['class'].values
```

```
X = [d.split() for d in data['text'].tolist()]
```

```
type(X)
```

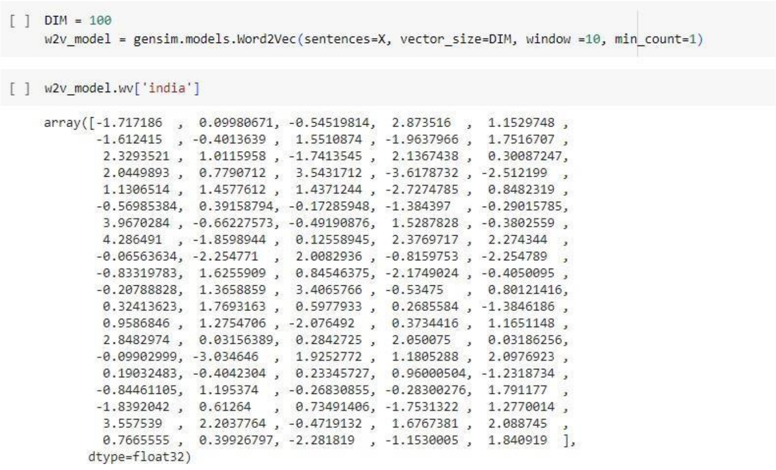
```
list
```

```
type(X[0])
```

```
list
```

```
print(X[0])
```

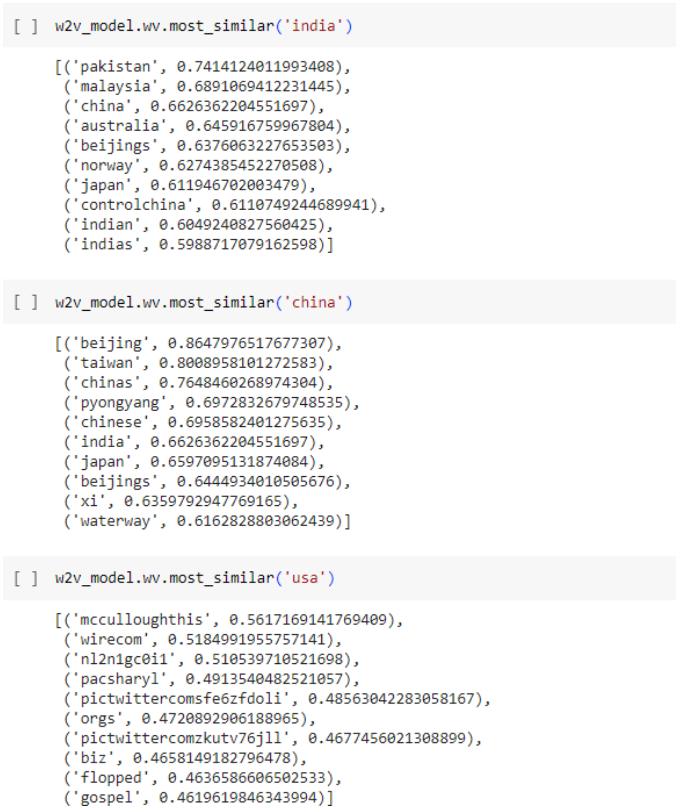
```
['as', 'us', 'budget', 'fight', 'looms', 'republicans', 'flip', 'their', 'fiscal', 'script', 'the', 'head', 'of', 'a', 'conservative', 'republican', 'faction', 'in', 'the', 'us', 'congress', 'who', 'voted', 'this', 'month', 'for', 'a', 'hu
```



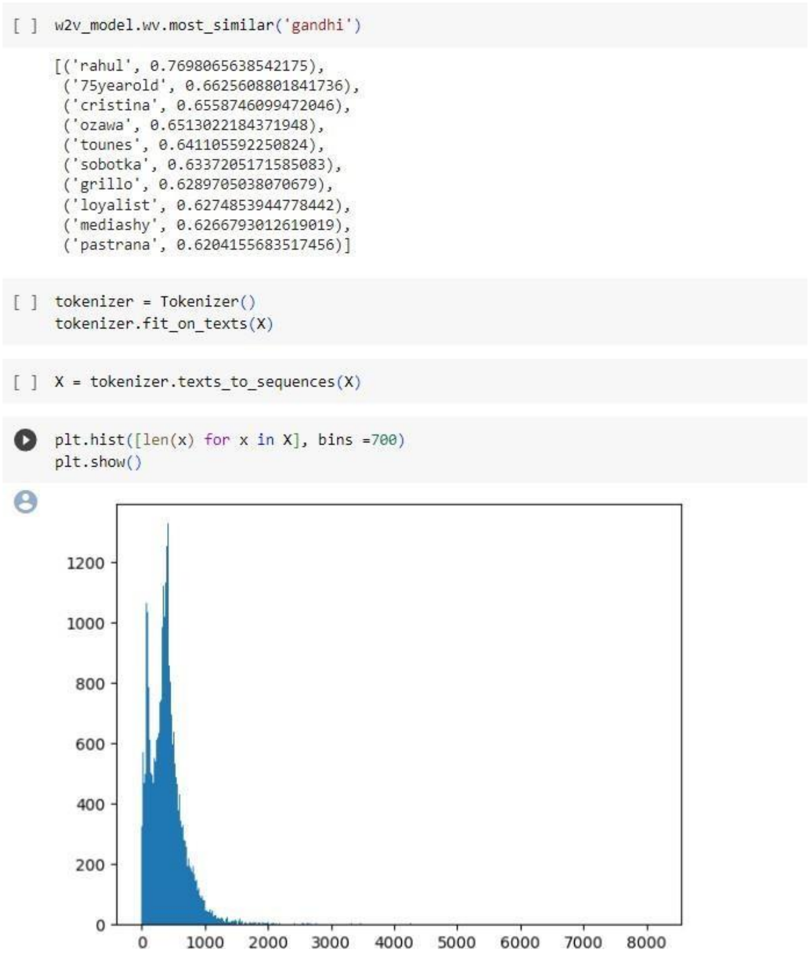
```
DIM = 100
w2v_model = gensim.models.Word2Vec(sentences=X, vector_size=DIM, window =10, min_count=1)
```

```
w2v_model.wv['india']
```

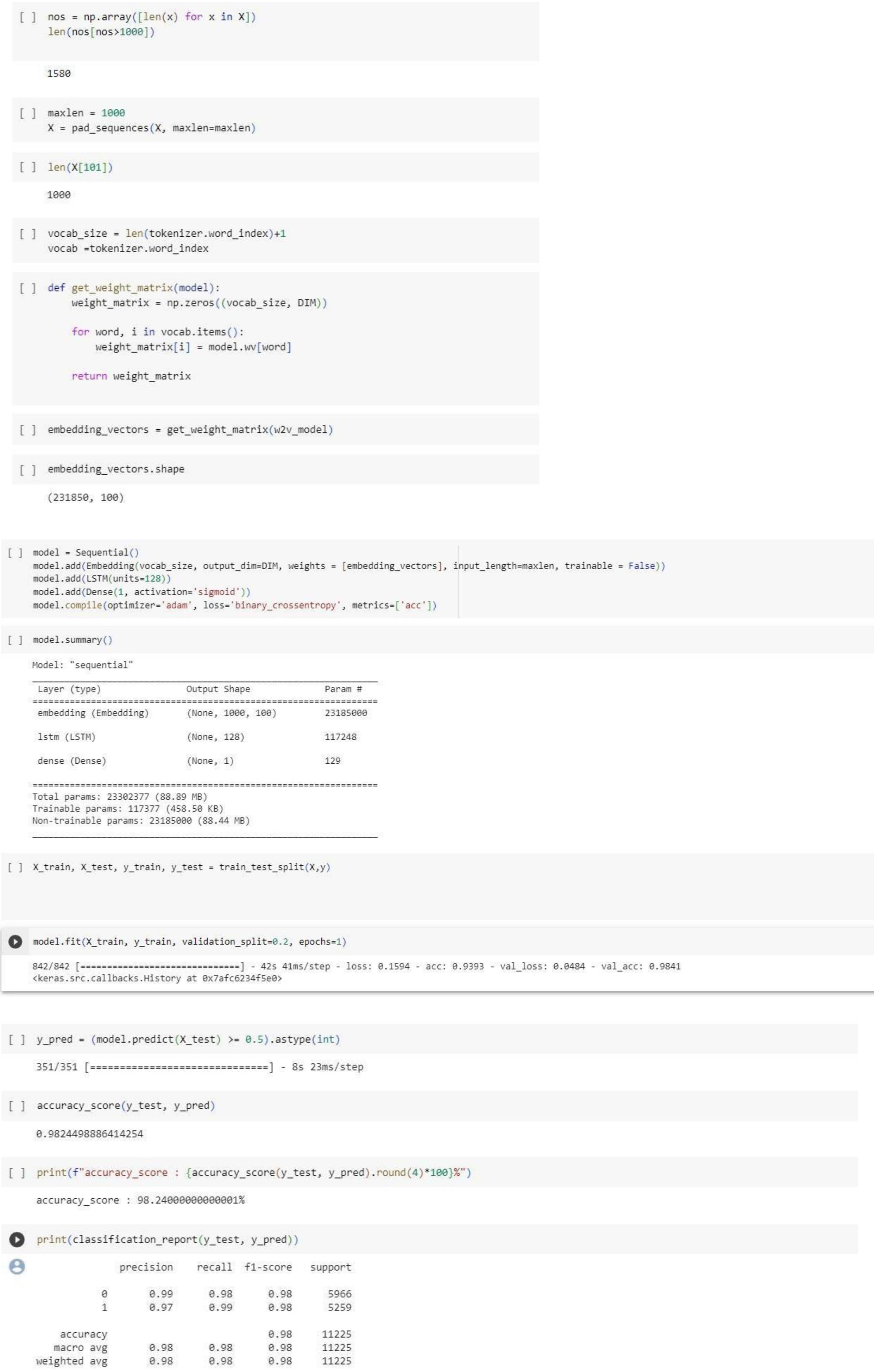
```
array([-1.717186  ,  0.09980671, -0.54519814,  2.873516  ,  1.1529748 ,
       -1.612415  , -0.4013639 ,  1.5510874 , -1.9637966 ,  1.7516707 ,
        2.3293521 ,  1.0115958 , -1.7413545 ,  2.1367438 ,  0.30087247,
        2.0449893 ,  0.7790712 ,  3.5431712 , -3.6178732 , -2.512199  ,
        1.1306514 ,  1.4577612 ,  1.4371244 , -2.7274785 ,  0.8482319 ,
       -0.56985384,  0.39158794, -0.17285948, -1.384397  , -0.29015785,
        3.9670284 , -0.66227573, -0.49190876,  1.5287828 , -0.3802559 ,
        4.286491  , -1.8598944 ,  0.12558945,  2.3769717 ,  2.274344  ,
       -0.06563634, -2.254771  ,  2.0082936 , -0.8159753 , -2.254789  ,
       -0.83319783,  1.6255909 ,  0.84546375, -2.1749024 , -0.4050095 ,
       -0.20788828,  1.3658859 ,  3.4065766 , -0.53475   ,  0.80121416,
        0.32413623,  1.7693163 ,  0.5977933 ,  0.2685584 , -1.3846186 ,
        0.9586846 ,  1.2754706 , -2.076492  ,  0.3734416 ,  1.1651148 ,
        2.8482974 ,  0.03156389,  0.2842725 ,  2.050075  ,  0.03186256,
       -0.09902999, -3.034646  ,  1.9252772 ,  1.1805288 ,  2.0976923 ,
        0.19032483, -0.4042304 ,  0.23345727,  0.96000504, -1.2318734 ,
       -0.84461105,  1.195374  , -0.26830855, -0.28300276,  1.791177  ,
       -1.8392042 ,  0.61264   ,  0.73491406, -1.7531322 ,  1.2770014 ,
        3.557539  ,  2.2037764 , -0.4719132 ,  1.6767381 ,  2.088745  ,
        0.7665555 ,  0.39926797, -2.281819  , -1.1530005 ,  1.840919  ],
      dtype=float32)
```



```
[ ]  w2v_model.wv.most_similar('india')

     [('pakistan', 0.7414124011993408),
      ('malaysia', 0.6891069412231445),
      ('china', 0.6626362204551697),
      ('australia', 0.645916759967804),
      ('beijings', 0.6376063227653503),
      ('norway', 0.6274385452270508),
      ('japan', 0.611946702003479),
      ('controlchina', 0.6110749244689941),
      ('indian', 0.6049240827560425),
      ('indias', 0.5988717079162598)]

[ ]  w2v_model.wv.most_similar('china')

     [('beijing', 0.8647976517677307),
      ('taiwan', 0.8008958101272583),
      ('chinas', 0.7648460268974304),
      ('pyongyang', 0.6972832679748535),
      ('chinese', 0.6958582401275635),
      ('india', 0.6626362204551697),
      ('japan', 0.6597095131874084),
      ('beijings', 0.6444934010505676),
      ('xi', 0.6359792947769165),
      ('waterway', 0.6162828803062439)]

[ ]  w2v_model.wv.most_similar('usa')

     [('mcculloughthis', 0.5617169141769409),
      ('wirecom', 0.5184991955757141),
      ('nl2n1gc0i1', 0.510539710521698),
      ('pacsharyl', 0.4913540482521057),
      ('pictwittercomsfe6zfdoli', 0.48563042283058167),
      ('orgs', 0.4720892906188965),
      ('pictwittercomzkutv76jll', 0.4677456021308899),
      ('biz', 0.4658149182796478),
      ('flopped', 0.4636586606502533),
      ('gospel', 0.4619619846343994)]
```



```
[ ]  w2v_model.wv.most_similar('gandhi')

     [('rahul', 0.7698065638542175),
      ('75yearold', 0.6625608801841736),
      ('cristina', 0.6558746099472046),
      ('ozawa', 0.6513022184371948),
      ('tounes', 0.641105592250824),
      ('sobotka', 0.6337205171585083),
      ('grillo', 0.6289705038070679),
      ('loyalist', 0.6274853944778442),
      ('mediashy', 0.6266793012619019),
      ('pastrana', 0.6204155683517456)]

[ ]  tokenizer = Tokenizer()
     tokenizer.fit_on_texts(X)

[ ]  X = tokenizer.texts_to_sequences(X)

     plt.hist([len(x) for x in X], bins =700)
     plt.show()
```



```
nos = np.array([len(x) for x in X])
len(nos[nos>1000])
```

```
1580
```

```
maxlen = 1000
X = pad_sequences(X, maxlen=maxlen)
```

```
len(X[101])
```

```
1000
```

```
vocab_size = len(tokenizer.word_index)+1
vocab =tokenizer.word_index
```

```
def get_weight_matrix(model):
    weight_matrix = np.zeros((vocab_size, DIM))

    for word, i in vocab.items():
        weight_matrix[i] = model.wv[word]

    return weight_matrix
```

```
embedding_vectors = get_weight_matrix(w2v_model)
```

```
embedding_vectors.shape
```

```
(231850, 100)
```

```
model = Sequential()
model.add(Embedding(vocab_size, output_dim=DIM, weights = [embedding_vectors], input_length=maxlen, trainable = False))
model.add(LSTM(units=128))
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['acc'])
```

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 1000, 100)         23185000

 lstm (LSTM)                 (None, 128)               117248

 dense (Dense)               (None, 1)                 129

=================================================================
Total params: 23302377 (88.89 MB)
Trainable params: 117377 (458.50 KB)
Non-trainable params: 23185000 (88.44 MB)
_________________________________________________________________
```

```
X_train, X_test, y_train, y_test = train_test_split(X,y)
```

```
model.fit(X_train, y_train, validation_split=0.2, epochs=1)
```

```
842/842 [==============================] - 42s 41ms/step - loss: 0.1594 - acc: 0.9393 - val_loss: 0.0484 - val_acc: 0.9841
<keras.src.callbacks.History at 0x7afc6234f5e0>
```

```
y_pred = (model.predict(X_test) >= 0.5).astype(int)
```

```
351/351 [==============================] - 8s 23ms/step
```

```
accuracy_score(y_test, y_pred)
```

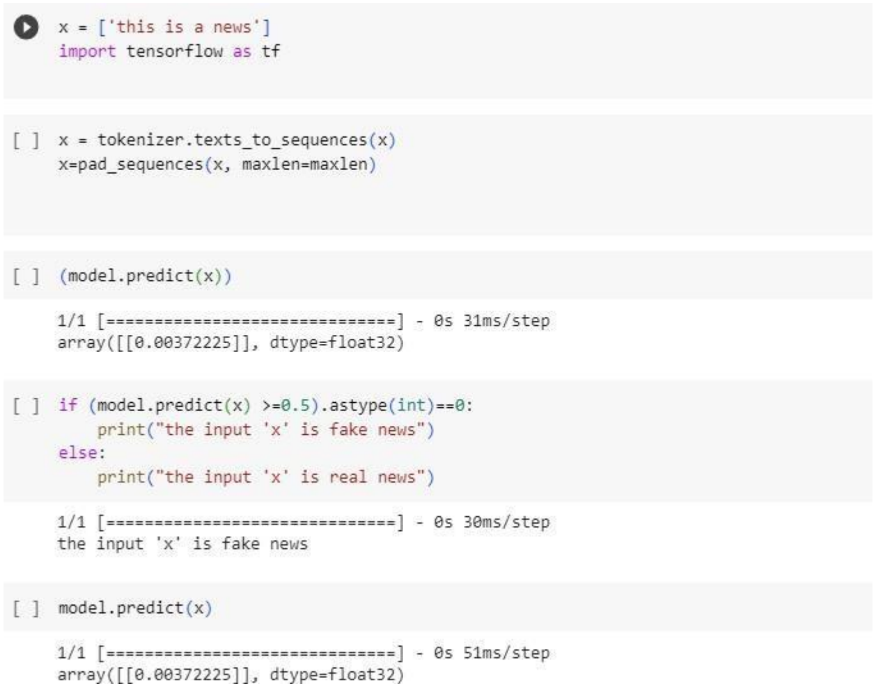
```
0.9824498886414254
```

```
print(f"accuracy_score : {accuracy_score(y_test, y_pred).round(4)*100}%")
```

```
accuracy_score : 98.24000000000001%
```

```
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.99      0.98      0.98      5966
           1       0.97      0.99      0.98      5259

    accuracy                           0.98     11225
   macro avg       0.98      0.98      0.98     11225
weighted avg       0.98      0.98      0.98     11225
```



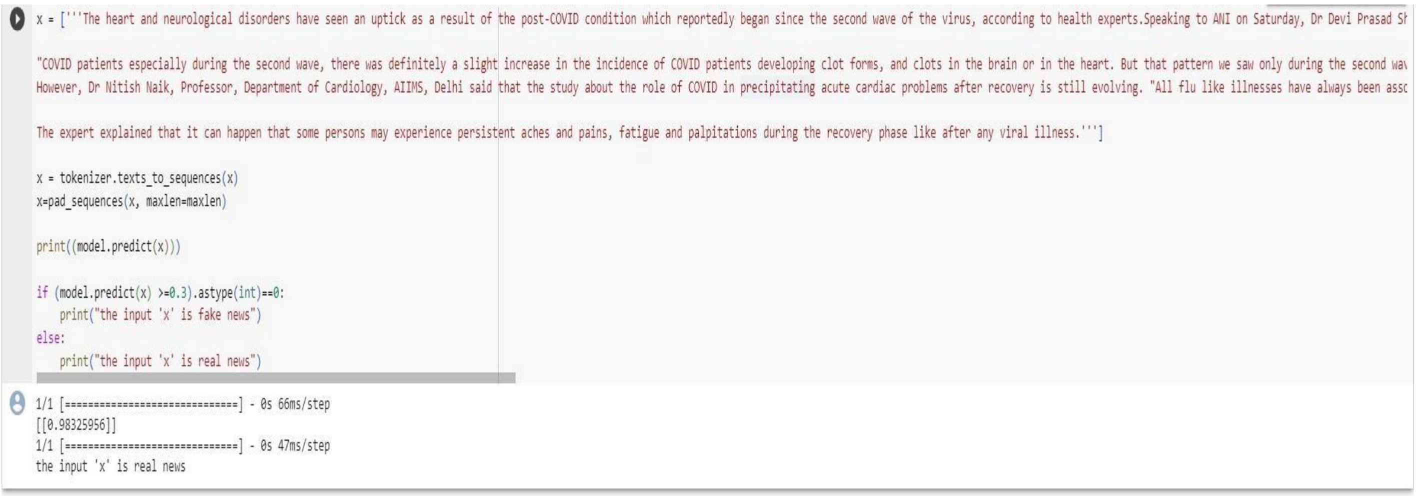
```
x = ['this is a news']
import tensorflow as tf
```

```
[ ] x = tokenizer.texts_to_sequences(x)
    x=pad_sequences(x, maxlen=maxlen)
```

```
[ ] (model.predict(x))

    1/1 [==============================] - 0s 31ms/step
    array([[0.00372225]], dtype=float32)
```

```
[ ] if (model.predict(x) >=0.5).astype(int)==0:
        print("the input 'x' is fake news")
    else:
        print("the input 'x' is real news")

    1/1 [==============================] - 0s 30ms/step
    the input 'x' is fake news
```

```
[ ] model.predict(x)

    1/1 [==============================] - 0s 51ms/step
    array([[0.00372225]], dtype=float32)
```



```
x = ['''The heart and neurological disorders have seen an uptick as a result of the post-COVID condition which reportedly began since the second wave of the virus, according to health experts.Speaking to ANI on Saturday, Dr Devi Prasad Sh

"COVID patients especially during the second wave, there was definitely a slight increase in the incidence of COVID patients developing clot forms, and clots in the brain or in the heart. But that pattern we saw only during the second wav
However, Dr Nitish Naik, Professor, Department of Cardiology, AIIMS, Delhi said that the study about the role of COVID in precipitating acute cardiac problems after recovery is still evolving. "All flu like illnesses have always been assc

The expert explained that it can happen that some persons may experience persistent aches and pains, fatigue and palpitations during the recovery phase like after any viral illness.''']

x = tokenizer.texts_to_sequences(x)
x=pad_sequences(x, maxlen=maxlen)

print((model.predict(x)))

if (model.predict(x) >=0.3).astype(int)==0:
    print("the input 'x' is fake news")
else:
    print("the input 'x' is real news")

1/1 [==============================] - 0s 66ms/step
[[0.98325956]]
1/1 [==============================] - 0s 47ms/step
the input 'x' is real news
```

Conclusion:

In conclusion, utilizing Natural Language Processing (NLP) techniques for fake news detection has proven to be a significant advancement in combating misinformation. The model developed demonstrates the potential of machine learning in identifying deceptive content, contributing to the ongoing efforts to maintain the integrity of information online. By leveraging NLP algorithms, the accuracy and efficiency of fake news detection have been greatly enhanced, empowering users to make informed decisions and fostering a more reliable digital information ecosystem. As we move forward, continued research and development in this field will play a pivotal role in ensuring the authenticity and trustworthiness of online content, thereby promoting a healthier and more informed society.