Fake News Detection Using NLP

INTRODUCTION :

In today's digital age, the spread of information has become faster and more widespread than ever before. While this has many advantages, it also opens the door to a growing problem: the proliferation of fake news. False or misleading information can easily go viral, causing harm to individuals, organizations, and society at large. In response to this, the field of Natural Language Processing (NLP) has emerged as a powerful tool in the battle against fake news.

NLP is a subfield of artificial intelligence that focuses on the interaction between computers and human language. It equips machines with the ability to understand, interpret, and generate human language, making it an essential technology in the fight against the dissemination of false information. Detecting fake news using NLP techniques involves the application of computational algorithms to automatically analyze and verify the authenticity of textual content, thereby helping users make more informed decisions about the information they encounter.

This interdisciplinary approach combines the power of linguistics, data science, and machine learning to identify patterns, anomalies, and inconsistencies within text. By leveraging NLP, we can scrutinize news articles, social media posts, and other textual data to differentiate between trustworthy information and fabricated or misleading content. The goal is to develop robust, automated systems that can identify and flag suspicious content, ultimately curbing the impact of fake news on public perception, decision-making, and societal stability.

This comprehensive guide to fake news detection using NLP explores the various techniques, methodologies, and tools employed to combat the spread of misinformation. It delves into the challenges and complexities of identifying fake news in an age where information is abundant and easily manipulated. Throughout this journey, we will discover the key NLP approaches and strategies that have been developed to confront this modern dilemma, shedding light on the innovative solutions that researchers and organizations are actively pursuing.

As the battle against fake news intensifies, NLP stands at the forefront of the fight, providing a glimmer of hope in an era of information overload. By understanding the foundations of NLP-based fake news detection, we can better equip ourselves to critically assess the information we encounter and ensure a more truthful and reliable digital landscape for all.

In our hyperconnected world, the information landscape is evolving at an unprecedented pace, fueled by the widespread adoption of digital media and social platforms. While this connectivity offers numerous advantages, it also brings to the forefront a pressing and pernicious issue: the proliferation of fake news. False or deceptive information, masquerading as credible news, can swiftly circulate, causing confusion, sowing discord, and even influencing public opinion and policy decisions. In response to this growing problem, the application of Natural Language Processing (NLP) techniques has emerged as a formidable weapon in the ongoing battle against fake news.

NLP, a subfield of artificial intelligence (AI), centers on the interaction between computers and human language. It equips machines with the ability to understand, interpret, and generate human language, making it an invaluable asset in the effort to combat the dissemination of disinformation. Fake news detection using NLP involves the utilization of computational algorithms to automatically scrutinize, verify, and classify textual content, enabling users to make more informed judgments about the information they encounter.

This multidisciplinary approach amalgamates the power of linguistics, data science, and machine learning, offering the capacity to identify patterns, anomalies, and inconsistencies within text. By harnessing NLP, we can meticulously analyze news articles, social media posts, and a variety of textual data sources to differentiate between credible information and fabricated or misleading content. The ultimate objective is to develop robust, automated systems capable of recognizing and flagging suspicious content, thereby mitigating the impact of fake news on public perception, decision-making, and societal harmony.

This comprehensive exploration of fake news detection using NLP traverses the diverse techniques, methodologies, and tools that have been devised to confront the menace of misinformation. It delves into the intricacies and challenges of identifying fake news in an era where information is abundant and easily manipulated. Throughout this journey, we will uncover the fundamental NLP approaches and strategies that researchers and organizations are employing to address this contemporary dilemma. Together, we will shed light on the innovative solutions being actively pursued and gain a deeper understanding of how NLP can act as a vital instrument in the fight against the spread of fake news.

As the battle against fake news intensifies, NLP stands at the forefront, offering a beacon of hope in an era of information overload. By grasping the foundations of NLP-based fake news detection, we empower ourselves to critically assess the information we encounter and work toward a more truthful, reliable, and transparent digital landscape for all. In the pages that follow, we embark on a journey to unravel the complexities of this vital endeavor, aiming to equip readers with the knowledge and tools necessary to discern fact from fiction in an age of information abundance and uncertainty.

Module Title: Fake News Detection Using NLP

1. Data Collection:

Gather a diverse dataset of news articles, social media posts, or textual data containing both real and fake news. Datasets like Snopes, Politifact, or Twitter's API can be valuable sources.

2. Data Preprocessing:

Text Cleaning: Remove HTML tags, special characters, and irrelevant content.

Tokenization: Split text into words or subwords.

Stopword Removal: Eliminate common words (e.g., "the," "and") that don't carry significant information.

Lemmatization or Stemming: Reduce words to their base form.

Vectorization: Transform text into numerical features using techniques like TF-IDF or Word Embeddings (e.g., Word2Vec, GloVe).

3. Feature Engineering:

Extract relevant features, such as n-grams, sentiment analysis, and part-of-speech tags, which can enhance the model's performance.

4. Model Selection:

Choose a machine learning or deep learning model suitable for NLP tasks. Common choices include:

Naive Bayes Classifier

Logistic Regression

Random Forest

Support Vector Machine (SVM)

Recurrent Neural Networks (RNN)

Convolutional Neural Networks (CNN)

Transformer-based models (e.g., BERT, GPT)

5. Model Training:

Split the dataset into training, validation, and testing sets.

Train the selected model on the training data using the preprocessed features.

Fine-tune hyperparameters using the validation set to optimize performance.

6. Model Evaluation:

Assess the model's performance using various metrics such as accuracy, precision, recall, F1-score, and ROC-AUC.

Conduct cross-validation to ensure robustness.

Visualize results, including confusion matrices and ROC curves.

7. Model Interpretability :

Use techniques like LIME (Local Interpretable Model-agnostic Explanations) or SHAP (SHapley Additive exPlanations) to understand why the model makes specific predictions.

8. Deployment:

Create a user-friendly interface for users to input text and receive predictions.

Deploy the model on a web server, cloud platform, or as a desktop application.

9. Continuous Monitoring and Maintenance:

Periodically retrain the model with fresh data to adapt to evolving fake news tactics.

Maintain an update pipeline for the NLP model, including monitoring for model drift and performance degradation.

10. User Education:

Provide educational materials and guidelines to help users understand how to use the tool effectively and interpret the results.

11. Ethical Considerations:

Ensure that the module adheres to ethical guidelines, respects privacy, and avoids biases in decision-making.

12. Scaling and Optimization :

For larger-scale applications, consider distributed computing and optimization techniques to handle a higher volume of data and requests.

13. Feedback Mechanism:

Implement a feedback loop to collect user feedback and continuously improve the module.

Creating a Fake News Detection module using NLP is an ongoing process that requires a multidisciplinary approach, combining expertise in NLP, machine learning, and domain knowledge. Regular updates and improvements will help keep the system effective in combating the evolving challenges of fake news dissemination.

REAL AND FACK NEWS DATA SET

Module 1: Data Collection and Preprocessing

In Module 1 of our research project, we focus on the critical task of collecting and preprocessing a comprehensive dataset of news articles. This module encompasses the following key steps:

➢ Data Gathering: We employ web scraping techniques to acquire news articles from diverse online sources, including reputable news websites and social media platforms. This dataset comprises both real and potentially fake news articles to ensure a balanced representation.

➢ Data Cleaning: To enhance the quality of the collected data, we perform extensive data cleaning, including text normalization, removal of duplicates, and extraction of relevant features. Additionally, we assess and handle missing data points.

➢ Labeling: Human annotators are engaged to label the collected news articles as either "real" or "fake" based on factual accuracy and credibility. This ground truth labeling is crucial for supervised machine learning models.

Module 2: Feature Engineering and Selection

Module 2 focuses on feature engineering and selection techniques to extract meaningful information from the preprocessed dataset. The steps involved in this module are as follows:

➢ Text Representation: We explore various text representation methods such as TF-IDF (Term Frequency-Inverse Document Frequency) and word embeddings (e.g., Word2Vec, GloVe) to convert textual data into numerical features.

➢ Feature Extraction: We extract linguistic, semantic, and contextual features from the text, including n-grams, sentiment analysis scores, and topic modeling. These features are designed to capture patterns and nuances in news articles.

➢ Feature Selection: Employing feature selection algorithms, we identify and retain the most relevant features while eliminating noise. This helps improve the model's efficiency and interpretability.

Module 3: Machine Learning Models for Classification

Module 3 is dedicated to building and evaluating machine learning models for the classification of news articles into real and fake categories. This module encompasses the following steps:

➢ Model Selection: We experiment with a range of classification algorithms, including logistic

regression, random forests, and deep neural networks. Each model's performance is assessed using appropriate evaluation metrics such as accuracy, precision, recall, and F1score.

➢ Model Training and Tuning: The selected models are trained on the preprocessed dataset, and hyperparameter tuning is performed to optimize their performance.

➢ Cross-Validation: To ensure the generalizability of our models, we employ cross-validation techniques to assess their robustness and reliability.

Module 4: Evaluation and Conclusion

In the final module, we evaluate the effectiveness of our real and fake news detection models using an independent test dataset. We analyze the results, discuss insights, and draw conclusions regarding the performance and limitations of the models.

Overall, our research project aims to provide a comprehensive framework for real and fake news detection, from data collection to model evaluation. Through this systematic approach, we contribute to the ongoing efforts to combat misinformation and enhance the credibility of online news sources.

PROGRAM & OUTPUT :

IMPORT LIBRARIES

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import torch
import torch.nn as nn
import nltk
from tqdm import tqdm
```

```
import torchtext.data as data
import torch.optim as optim
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from torchtext.data import get_tokenizer
import seaborn as sns
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
import re
```

in [2]:

```
# CONFIG
TRUE_DATA_PATH = '/kaggle/input/fake-and-real-news-dataset/True.csv'
FALSE_DATA_PATH = '/kaggle/input/fake-and-real-news-dataset/Fake.csv'
```

LOAD DATA

In [3]:

```
true_df = pd.read_csv(TRUE_DATA_PATH)
false_df = pd.read_csv(FALSE_DATA_PATH)
```

In [4]:

```
true_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21417 entries, 0 to 21416
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   title    21417 non-null  object
 1   text     21417 non-null  object
 2   subject  21417 non-null  object
 3   date     21417 non-null  object
dtypes: object(4)
memory usage: 669.4+ KB
```

In [5]:

```
false_df.info();
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23481 entries, 0 to 23480
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   title    23481 non-null  object
 1   text     23481 non-null  object
 2   subject  23481 non-null  object
 3   date     23481 non-null  object
dtypes: object(4)
memory usage: 733.9+ KB
```

In [6]:

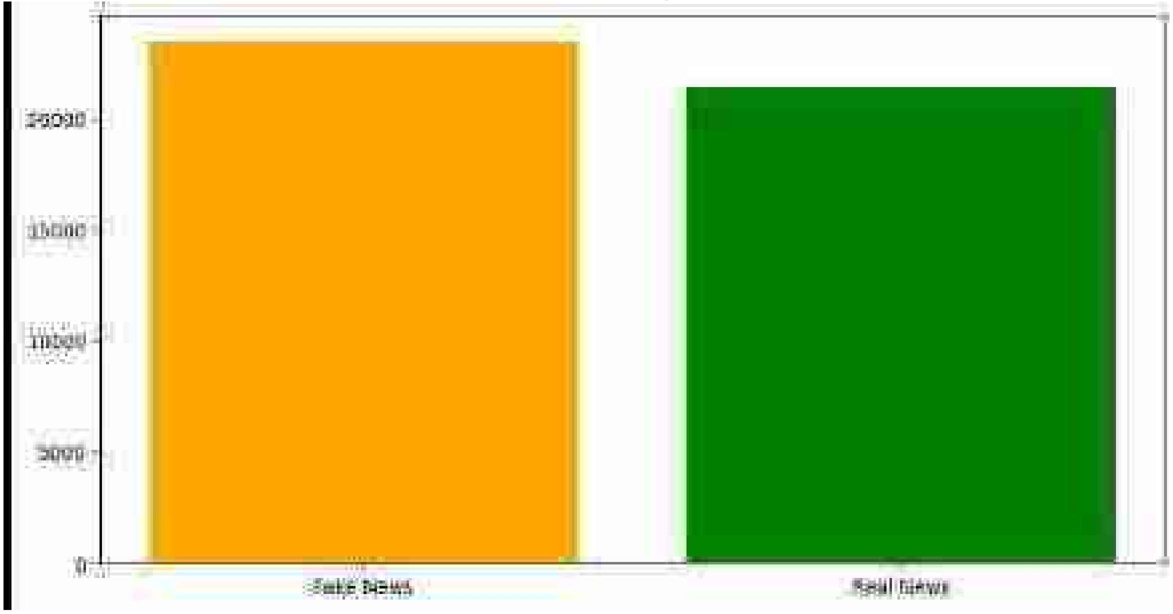
```
true_df['category'] = np.ones(len(true_df), dtype=int)
false_df['category'] = np.zeros(len(false_df), dtype=int)

true_df.head()
```

In [7]:

```
plt.figure(figsize=(10, 5))
plt.bar('Fake News', len(false_df), color='orange')
plt.bar('Real News', len(true_df), color='green')
```

out[7]:

```
<BarContainer object of 1 artists>
```



In [8]:

```
# Difference of the Fake and Real News
print(f'Difference between Fake and Real News: {len(false_df) - len(true_df)}')
```

```
Difference between Fake and Real News: 2064
```

In [9]:

```
# concat = merging datasets
news_df = pd.concat([true_df, false_df], axis=0)
news_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
Index: 44898 entries, 0 to 23480
Data columns (total 5 columns):
 #   Column    Non-Null Count  Data type
---  ------    --------------  -----
 0   title     44898 non-null  object
 1   text      44898 non-null  object
 2   subject   44898 non-null  object
 3   date      44898 non-null  object
 4   category  44898 non-null  int64
Data types: int64(1), object(4)
memory usage: 2.1+ MB
```

In [10]:

```
news_df = news_df.sample(frac=1)
news_df.head(5)
```

out[10]:

	Title	Text	subject	date	category
880	These Charts Show Why We're All Screwed Under...	During his presidential campaign, Donald Trump...	News	July 11, 2017	0
597	U.S. towns, cities fear taxpayer revolt if Rep...	WASHINGTON (Reuters) - From Pataskala, Ohio, t...	politicsNews	November 17, 2017	1
15813	JEB BUSH WANTS CONGRESS TO APPROVE AMNESTY And...	Jeb Bush just unofficially placed himself on t...	politics	Apr 17, 2015	0
15407	DEMOCRAT PLAN TO INFILTRATE TRADITIONALLY RED ...	The fundamental transformation of America El S...	politics	Jul 27, 2015	0
18289	NEW YORK TIMES REFUSES To Publish Op-Ed By Lif...	The NYT allegedly wouldn t run Alan Dershowitz...	left-news	Jul 20, 2017	0

In [11]:

```
news_df['subject'].value_counts()
```

out[11]:

```
subject
politicsNews       11272
worldnews          10145
News                9050
politics            6841
left-news           4459
Government News     1570
US_News              783
Middle-east          778
Name: count, dtype: int64
```

in[12]:

```
news_df = pd.get_dummies(news_df, columns=['subject'])
news_df.head()
```

in[13]:

```
news_df = news_df.drop('date', axis=1)
news_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 44898 entries, 880 to 7431
Data columns (total 11 columns):
 #   Column                   Non-Null Count  Dtype 
---  ------                   --------------  ----- 
 0   title                    44898 non-null  object
 1   text                     44898 non-null  object
 2   category                 44898 non-null  int64 
 3   subject_Government News  44898 non-null  bool  
 4   subject_Middle-east      44898 non-null  bool  
 5   subject_News             44898 non-null  bool  
 6   subject_US_News          44898 non-null  bool  
 7   subject_left-news        44898 non-null  bool  
 8   subject_politics         44898 non-null  bool  
 9   subject_politicsNews     44898 non-null  bool  
 10  subject_worldnews        44898 non-null  bool  
dtypes: bool(8), int64(1), object(2)
memory usage: 1.7+ MB
```

in[14]

```
import nltk
import subprocess
import nltk
import subprocess

try:
    nltk.data.find('wordnet.zip')
except:
    nltk.download( download_dir='wornet')
    command = copora wornet’
    subprocess.run(command.split())
    nltk.data.path.append('working')

from nltk.corpus import wordnet
```

In[15]:

```



from nltk.corpus import wordnet

new_text = []
pattern = "[^a-zA-Z]"

lemma = nltk.WordNetLemmatizer()

for txt in tqdm(news_df.text):

    txt = re.sub(pattern," ",txt)
    txt = txt.lower() # Lowering
    txt = nltk.word_tokenize(txt)
    txt = [lemma.lemmatize(word) for word in txt]
    txt = " ".join(txt)
    new_text.append(txt)

new_text[0]
```

```
100%|██████████| 44898/44898 [05:21<00:00, 139.84it/s]
```

Out[15]:

```
'during his presidential campaign donald trump constantly made reference to
repealing and replacing the disaster that is obamacare and democrat collectively
shuddered we all knew that nothing good could come of this now after six month in
office despite discovering that nobody knew healthcare could be so difficult
president trump is about to deliver on his campaign promise a the senate return
from a one week recess to get back to the task at hand trying to come to an
agreement on their new healthcare bill known a the better care reconciliation act
bcra one that they have predominately kept the public in the dark about thing are
```

looking bleak however a even the republican party remains divided on a bill that is not only going to raise out of pocket cost and restrict access to service for many but also cause ten of million to lose their health insurance completely over the coming decade these chart might be able to put the entire healthcare debacle into perspective we all know the short term medicaid cut are going to suck the new health care bill will save a ton of money but roughly a quarter of those saving or approximately billion over the next decade come from cut to medicaid the result is that million le 'during his presidential campaign donald trump constantly made reference to repealing and replacing the disaster that is obamacare and democrat collectively shuddered we all knew that nothing good could come of this now after six month in office despite discovering that nobody knew healthcare could be so difficult president trump is about to deliver on his campaign promise a the senate return from a one week recess to get back to the task at hand trying to come to an agreement on their new healthcare bill known a the better care reconciliation act bcra one that they have predominately kept the public in the dark about thing are looking bleak however a even the republican party remains divided on a bill that is not only going to raise out of pocket cost and restrict access to service for many but also cause ten of million to lose their health insurance completely over the coming decade these chart might be able to put the entire healthcare debacle into perspective we all know the short term medicaid cut are going to suck the new health care bill will save a ton of money but roughly a quarter of those saving or approximately billion over the next decade come from cut to medicaid the result is that million le including those whose overwhelming majority voted for trump yes the new bill will drive up the uninsured rate by at least and even up to in every state by a new study by the urban institute found the older and poorer you are the more you will be paying for insurance premium if an analysis by the center for budget and policy priority is to be believed health insurance premium are going to go through the roof but those hit the worst will be older american the older middle class will be hit pretty hard too a their tax credit will go through the floor the center for budget and policy priority analysis also found that the tax credit that are available to help older people in the individual market afford health insurance are going to do just the opposite and plummet even employer plan aren t immune the gop s new bill cut to medicaid and individual market subsidy have given the million american that receive their health insurance through their employer a false sense of security but they re not safe either not only will the new legislation bring back annual and lifetime limit in employer plan a well a end penalty for company that don t provide health insurance to their worker but it will also allow employer to shift much of the cost of copays deductible and coinsurance onto their worker the center for american progress calculated how many will feel the crunch hospital are going to feel the crunch a well hospital aren t happy with the new bill and it is easy to see why when you consider it will cause a large spike in uncompensated care for hospital across all state finally the new bill will cause massive job loss particularly in the health care sector by more than million job will be lost a a direct result of the bcra go by the result of a report by the commonwealth fund and george washington university in fact the report go a far a to say that every state except hawaii would have fewer job and a weaker economy however it s not just health care employment that will be affected but also retail and construction a well so if you thought this latest rewrite of the gop s health care legislation didn t affect you you more than likely thought wrongly even if it isn t your health care that is directly affected chance are you will still feel the ripple effect of the bill on the

```
economy both on a state and national level featured image via drew angerer getty
image'.
```

In[16]:

```
new_title = []
for txt in tqdm(news_df.title):

    txt = re.sub(pattern," ",txt)
    txt = txt.lower() # Lowering
    txt = nltk.word_tokenize(txt)
    txt = [lemma.lemmatize(word) for word in txt]
    txt = " ".join(txt)
    new_title.append(txt)
new_title[0]
```

```
100%|██████████| 44898/44898 [00:15<00:00, 2941.90it/s]
```

Out [16]:

```
'these chart show why we re all screwed under the gop health care bill'
```

In[17]:

```
from sklearn.feature_extraction.text import CountVectorizer

vectorizer_title = CountVectorizer(stop_words="english",max_features=1000)
vectorizer_text = CountVectorizer(stop_words="english",max_features=4000)

title_matrix = vectorizer_title.fit_transform(new_title).toarray()
text_matrix = vectorizer_text.fit_transform(new_text).toarray()

print("Finished")
```

```
Finished
```

In[18]

```
news_df.head(5)
```

in[19]:

```
news_df.drop(['title', 'text'], axis=1, inplace=True)
news_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 44898 entries, 880 to 7431
Data columns (total 9 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   category                 44898 non-null  int64
 1   subject_Government News  44898 non-null  bool
 2   subject_Middle-east      44898 non-null  bool
 3   subject_News             44898 non-null  bool
 4   subject_US_News          44898 non-null  bool
 5   subject_left-news        44898 non-null  bool
 6   subject_politics         44898 non-null  bool
 7   subject_politicsNews     44898 non-null  bool
 8   subject_worldnews        44898 non-null  bool
dtypes: bool(8), int64(1)
memory usage: 1.0 MB
```

in [20]:

```
print(news_df.shape)
print(title_matrix.shape)
print(text_matrix.shape)
```

```
(44898, 9)
(44898, 1000)
(44898, 4000)
```

In[21]:

```
X = np.concatenate((np.array(news_df.drop('category', axis=1)), title_matrix,
                     text_matrix), axis=1)

y = news_df.category
```

```
(44898, 5008)
(44898,)
```

in [23]:

```
X_train, X_test, y_train, y_test = train_test_split(X, np.array(y),
                                                    test_size=0.25,
                                                    random_state=42)

print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(33673, 5008)
(11225, 5008)
(33673,)
(11225,)
```

BUILDING MODEL

In[24]:

INTRODUCTION

Hello and welcome to my first NLP project, identifying fake and real news data using pytorch and nltk.

IMPORT LIBRARIES

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import torch
import torch.nn as nn
import nltk
from tqdm import tqdm
import torchtext.data as data
import torch.optim as optim
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from torchtext.data import get_tokenizer
import seaborn as sns
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
import re
```

In [2]:

```
# CONFIG
TRUE_DATA_PATH = ‘input fake-and-real-news-dataset/True.csv'
FALSE_DATA_PATH = ‘output fake-and-real-news-dataset/True.csv'
```

LOAD DATA

In [3]:

```
true_df = pd.read_csv(TRUE_DATA_PATH)
false_df = pd.read_csv(FALSE_DATA_PATH)
```

In [4]:

```
true_df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21417 entries, 0 to 21416
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   title    21417 non-null  object
 1   text     21417 non-null  object
 2   subject  21417 non-null  object
 3   date     21417 non-null  object
dtypes: object(4)
memory usage: 669.4+ KB
```

In [5]:

```
false_df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23481 entries, 0 to 23480
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   title    23481 non-null  object
 1   text     23481 non-null  object
 2   subject  23481 non-null  object
 3   date     23481 non-null  object
dtypes: object(4)
memory usage: 733.9+ KB
```

In [6]:

```
true_df['category'] = np.ones(len(true_df), dtype=int)
false_df['category'] = np.zeros(len(false_df), dtype=int)

true_df.head()
```

Out[6]:

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

In [7]:

```
plt.figure(figsize=(10, 5))
plt.bar('Fake News', len(false_df), color='orange')
plt.bar('Real News', len(true_df), color='green')
```

Out[7]:

```
<BarContainer object of 1 artists>
```

In [8]:

```
# Difference of the Fake and Real News
print(f'Difference between Fake and Real News: {len(false_df) - len(true_df)}')
Difference between Fake and Real News: 2064
```

In [9]:

```
# concat = merging datasets
news_df = pd.concat([true_df, false_df], axis=0)
news_df.info()
<class 'pandas.core.frame.DataFrame'>
Index: 44898 entries, 0 to 23480
Data columns (total 5 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   title     44898 non-null  object
 1   text      44898 non-null  object
 2   subject   44898 non-null  object
 3   date      44898 non-null  object
 4   category  44898 non-null  int64 
dtypes: int64(1), object(4)
memory usage: 2.1+ MB
```

In [10]:

```
news_df = news_df.sample(frac=1)
news_df.head(5)
```

Out[10]:

	Title	Text	subject	date	category
880	These Charts Show Why We're All Screwed Under...	During his presidential campaign, Donald Trump...	News	July 11, 2017	0
597	U.S. towns, cities fear taxpayer revolt if Rep...	WASHINGTON (Reuters) - From Pataskala, Ohio, t...	politicsNews	November 17, 2017	1
15813	JEB BUSH WANTS CONGRESS TO APPROVE AMNESTY And...	Jeb Bush just unofficially placed himself on t...	politics	Apr 17, 2015	0

	Title	Text	subject	date	category
15407	DEMOCRAT PLAN TO INFILTRATE TRADITIONALLY RED ...	The fundamental transformation of America El S...	politics	Jul 27, 2015	0
18289	NEW YORK TIMES REFUSES To Publish Op-Ed By Lif...	The NYT allegedly wouldn t run Alan Dershowitz...	left-news	Jul 20, 2017	0

In [11]:

```
news_df['subject'].value_counts()
```

Out[11]:

```
subject
politicsNews       11272
worldnews          10145
News                9050
politics            6841
left-news           4459
Government News     1570
US_News              783
Middle-east          778
Name: count, dtype: int64
```

In [12]:

```
news_df = pd.get_dummies(news_df, columns=['subject'])
news_df.head()
```

Out[12]:

	title	Text	Date	category	subject_Government News	subject_Middle-east	subject_News	subject_US_News	subject_left-news	subject_politics	subject_politicsNews	subject_worldnews
880	These Charts Show Why We’re All Screw	During his presidential campaign, Donal	July 11, 2017	0	False	False	True	False	False	False	False	False

	title	Text	Date	category	subject_Government News	subject_Middle-east	subject_News	subject_US_News	subject_left-news	subject_politics	subject_politicsNews	subject_worldnews
	ed Under...	d Trump ...										
597	U.S. towns, cities fear taxpayer revolt if Rep...	WASHINGTON (Reuters) - From Pataskala, Ohio, t...	November 17, 2017	1	False	False	False	False	False	False	True	False
15813	JEB BUSH WANTS CONGRESS TO APPROVE AMNESTY And...	Jeb Bush just unofficially placed himself on t...	Apr 17, 2015	0	False	False	False	False	False	True	False	False
15407	DEMOCRAT PLAN TO INFILTRATE TRADITIONALLY	The fundamental transformation of America El	Jul 27, 2015	0	False	False	False	False	False	True	False	False

	title	Text	Date	category	subject_Government News	subject_Middle-east	subject_News	subject_US_News	subject_left-news	subject_politics	subject_politicsNews	subject_worldnews
	RED ...	S...										
18289	NEW YORK TIMES REFUSES To Publish Op-Ed By Lif...	The NYT allegedly wouldn t run Alan Dershowitz...	Jul 20, 2017	0	False	False	False	False	True	False	False	False

In [13]:

```
news_df = news_df.drop('date', axis=1)
news_df.info()
<class 'pandas.core.frame.DataFrame'>
Index: 44898 entries, 880 to 7431
Data columns (total 11 columns):
 #   Column                   Non-Null Count  Dtype 
---  ------                   --------------  ----- 
 0   title                    44898 non-null  object
 1   text                     44898 non-null  object
 2   category                 44898 non-null  int64 
 3   subject_Government News  44898 non-null  bool  
 4   subject_Middle-east      44898 non-null  bool  
 5   subject_News             44898 non-null  bool  
 6   subject_US_News          44898 non-null  bool  
 7   subject_left-news        44898 non-null  bool  
 8   subject_politics         44898 non-null  bool  
 9   subject_politicsNews     44898 non-null  bool  
 10  subject_worldnews        44898 non-null  bool  
dtypes: bool(8), int64(1), object(2)
memory usage: 1.7+ MB
```

In [14]:

```
import nltk
import subprocess

# Download and unzip wordnet
try:
    nltk.data.find('wordnet.zip')
except:
    nltk.download('wordnet', download_dir='/kaggle/working/')
    command = "unzip /kaggle/working/corpora/wordnet.zip -d /kaggle/working/corpora"
    subprocess.run(command.split())
    nltk.data.path.append('/kaggle/working/')

# Now you can import the NLTK resources as usual
from nltk.corpus import wordnet
```

```
[nltk_data] Downloading package wordnet to /kaggle/working/...
Archive:  /kaggle/working/corpora/wordnet.zip
   creating: /kaggle/working/corpora/wordnet/
  inflating: /kaggle/working/corpora/wordnet/lexnames
  inflating: /kaggle/working/corpora/wordnet/data.verb
  inflating: /kaggle/working/corpora/wordnet/index.adv
  inflating: /kaggle/working/corpora/wordnet/adv.exc
  inflating: /kaggle/working/corpora/wordnet/index.verb
  inflating: /kaggle/working/corpora/wordnet/cntlist.rev
  inflating: /kaggle/working/corpora/wordnet/data.adj
  inflating: /kaggle/working/corpora/wordnet/index.adj
  inflating: /kaggle/working/corpora/wordnet/LICENSE
  inflating: /kaggle/working/corpora/wordnet/citation.bib
  inflating: /kaggle/working/corpora/wordnet/noun.exc
  inflating: /kaggle/working/corpora/wordnet/verb.exc
  inflating: /kaggle/working/corpora/wordnet/README
  inflating: /kaggle/working/corpora/wordnet/index.sense
  inflating: /kaggle/working/corpora/wordnet/data.noun
  inflating: /kaggle/working/corpora/wordnet/data.adv
  inflating: /kaggle/working/corpora/wordnet/index.noun
  inflating: /kaggle/working/corpora/wordnet/adj.exc
```

In [15]:

```
from nltk.corpus import wordnet

new_text = []
pattern = "[^a-zA-Z]"

lemma = nltk.WordNetLemmatizer()

for txt in tqdm(news_df.text):

    txt = re.sub(pattern," ",txt) # Cleaning
    txt = txt.lower() # Lowering
    txt = nltk.word_tokenize(txt) # Tokenizing
    txt = [lemma.lemmatize(word) for word in txt] # Lemmatizing
    txt = " ".join(txt)
    new_text.append(txt)

new_text[0]
```

```
100%|██████████| 44898/44898 [05:21<00:00, 139.84it/s]
```

Out[15]:

'during his presidential campaign donald trump constantly made reference to
repealing and replacing the disaster that is obamacare and democrat collectively
shuddered we all knew that nothing good could come of this now after six month in
office despite discovering that nobody knew healthcare could be so difficult
president trump is about to deliver on his campaign promise a the senate return
from a one week recess to get back to the task at hand trying to come to an
agreement on their new healthcare bill known a the better care reconciliation act
bcra one that they have predominately kept the public in the dark about thing are
looking bleak however a even the republican party remains divided on a bill that
is not only going to raise out of pocket cost and restrict access to service for
many but also cause ten of million to lose their health insurance completely over
the coming decade these chart might be able to put the entire healthcare debacle
into perspective we all know the short term medicaid cut are going to suck the
new health care bill will save a ton of money but roughly a quarter of those
saving or approximately billion over the next decade come from cut to medicaid
the result is that million le people will be enrolled in medicaid under the new
gop bill than compared to obamacare if you thought that wa bad the long term
effect are even worse the inflation rate for medicaid spending beginning in is
much slower affecting those who rely on it the most mainly child the disabled and
the elderly in fact by federal medicaid spending on child will be reduced by
almost a third and by a quarter for the disabled and the elderly when compared to
the current law according to an analysis by the health consulting firm avalere
health the percentage of those uninsured will rise in every single age bracket
that s right under the bcra million people will lose their insurance compared to
million under the version passed by the house and every single age group will be
affected according to an assessment by the congressional budget office it will
also rise in every state including those whose overwhelming majority voted for
trump yes the new bill will drive up the uninsured rate by at least and even up
to in every state by a new study by the urban institute found the older and
poorer you are the more you will be paying for insurance premium if an analysis
by the center for budget and policy priority is to be believed health insurance
premium are going to go through the roof but those hit the worst will be older
american the older middle class will be hit pretty hard too a their tax credit
will go through the floor the center for budget and policy priority analysis also
found that the tax credit that are available to help older people in the
individual market afford health insurance are going to do just the opposite and
plummet even employer plan aren t immune the gop s new bill cut to medicaid and
individual market subsidy have given the million american that receive their
health insurance through their employer a false sense of security but they re not
safe either not only will the new legislation bring back annual and lifetime
limit in employer plan a well a end penalty for company that don t provide health
insurance to their worker but it will also allow employer to shift much of the
cost of copays deductible and coinsurance onto their worker the center for
american progress calculated how many will feel the crunch hospital are going to
feel the crunch a well hospital aren t happy with the new bill and it is easy to
see why when you consider it will cause a large spike in uncompensated care for
hospital across all state finally the new bill will cause massive job loss
particularly in the health care sector by more than million job will be lost a a
direct result of the bcra go by the result of a report by the commonwealth fund
and george washington university in fact the report go a far a to say that every
state except hawaii would have fewer job and a weaker economy however it s not
just health care employment that will be affected but also retail and

```
construction a well so if you thought this latest rewrite of the gop s health
care legislation didn t affect you you more than likely thought wrongly even if
it isn t your health care that is directly affected chance are you will still
feel the ripple effect of the bill on the economy both on a state and national
level featured image via drew angerer getty image'
```

In [16]:

```
new_title = []
for txt in tqdm(news_df.title):

    txt = re.sub(pattern," ",txt) # Cleaning
    txt = txt.lower() # Lowering
    txt = nltk.word_tokenize(txt) # Tokenizing
    txt = [lemma.lemmatize(word) for word in txt] # Lemmatizing
    txt = " ".join(txt)
    new_title.append(txt)
new_title[0]
100%|██████████| 44898/44898 [00:15<00:00, 2941.90it/s]
```

Out[16]:

```
'these chart show why we re all screwed under the gop health care bill'
```

In [17]:

```
from sklearn.feature_extraction.text import CountVectorizer

vectorizer_title = CountVectorizer(stop_words="english",max_features=1000)
vectorizer_text = CountVectorizer(stop_words="english",max_features=4000)

title_matrix = vectorizer_title.fit_transform(new_title).toarray()
text_matrix = vectorizer_text.fit_transform(new_text).toarray()

print("Finished")
Finished
```

In [18]:

```
news_df.head(5)
```

Out[18]:

	title	Text	Category	subject_Government News	subject_Middle-east	subject_News	subject_US_News	subject_left-news	subject_politics	subject_politicsNews	subject_worldnews
880	These Charts Show Why We're All	During his presidential campaign,	0	False	False	True	False	False	False	False	False

	title	Text	Category	subject_Government News	subject_Middle-east	subject_News	subject_US_News	subject_left-news	subject_politics	subject_politicsNews	subject_worldnews
	Screwed Under...	Donald Trump...									
597	U.S. towns, cities fear taxpayer revolt if Rep...	WASHINGTON (Reuters) - From Pataskala, Ohio, t...	1	False	False	False	False	False	False	True	False
15813	JEB BUSH WANTS CONGRESS TO APPROVE AMNESTY And...	Jeb Bush just unofficially placed himself on t...	0	False	False	False	False	False	True	False	False
15407	DEMOCRAT PLAN TO INFILTRATE TRADITIONALLY RED ...	The fundamental transformation of America El S...	0	False	False	False	False	False	True	False	False

	title	Text	Category	subject_Government News	subject_Middle-east	subject_News	subject_US_News	subject_left-news	subject_politics	subject_politicsNews	subject_worldnews
18289	NEW YORK TIMES REFUSES To Publish Op-Ed By Lif...	The NYT allegedly wouldnt run Alan Dershowitz...	0	False	False	False	False	True	False	False	False

In [19]:

```
news_df.drop(['title', 'text'], axis=1, inplace=True)
news_df.info()
<class 'pandas.core.frame.DataFrame'>
Index: 44898 entries, 880 to 7431
Data columns (total 9 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   category                 44898 non-null  int64
 1   subject_Government News  44898 non-null  bool
 2   subject_Middle-east      44898 non-null  bool
 3   subject_News             44898 non-null  bool
 4   subject_US_News          44898 non-null  bool
 5   subject_left-news        44898 non-null  bool
 6   subject_politics         44898 non-null  bool
 7   subject_politicsNews     44898 non-null  bool
 8   subject_worldnews        44898 non-null  bool
dtypes: bool(8), int64(1)
memory usage: 1.0 MB
```

In [20]:

```
print(news_df.shape)
print(title_matrix.shape)
print(text_matrix.shape)
(44898, 9)
(44898, 1000)
(44898, 4000)
```

In [21]:

```
X = np.concatenate((np.array(news_df.drop('category', axis=1)), title_matrix,
                    text_matrix), axis=1)
```

```
y = news_df.category
```

In [22]:

```
print(X.shape)
print(y.shape)
```

```
(44898, 5008)
(44898,)
```

In [23]:

```
X_train, X_test, y_train, y_test = train_test_split(X, np.array(y),
                                                    test_size=0.25,
                                                    random_state=42)

print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(33673, 5008)
(11225, 5008)
(33673,)
(11225,)
```

BUILDING MODEL

In [24]:

```
import torch
import torch.nn as nn
import torch.nn.functional as F

class NewsClassifier(nn.Module):
    def __init__(self):
        super(NewsClassifier, self).__init__()

        # Fully connected layers
        self.linear1 = nn.Linear(5008, 2000)
        self.relu1 = nn.ReLU()
        self.linear2 = nn.Linear(2000, 500)
        self.relu2 = nn.ReLU()
        self.linear3 = nn.Linear(500, 100)
        self.relu3 = nn.ReLU()
        self.dropout = nn.Dropout(0.1)
        self.linear4 = nn.Linear(100, 20)
        self.relu4 = nn.ReLU()
        self.linear5 = nn.Linear(20, 2)

    def forward(self, x):
        # Fully connected layers
        out = self.linear1(x)
        out = self.relu1(out)
        out = self.linear2(out)
        out = self.relu2(out)
        out = self.linear3(out)
        out = self.relu3(out)
        out = self.dropout(out)
        out = self.linear4(out)
        out = self.relu4(out)
        out = self.linear5(out)

        return out
```

In [25]:

```
model = NewsClassifier()
optimizer = torch.optim.Adam(model.parameters(), lr=0.012)
criterion = nn.CrossEntropyLoss()
```

In [26]:

```
import torch
from tqdm import tqdm

X_train = torch.Tensor(X_train)
y_train = torch.Tensor(y_train).type(torch.LongTensor)

X_test = torch.Tensor(X_test)
y_test = torch.Tensor(y_test).type(torch.LongTensor)

EPOCHS = 30

for epoch in tqdm(range(EPOCHS)):
    optimizer.zero_grad()

    # Forward pass
    outputs = model(X_train)

    # Calculate loss
    loss = criterion(outputs, y_train)
    loss.backward()
    optimizer.step()

    # Calculate accuracy
    _, predicted = torch.max(outputs, 1)
    correct = (predicted == y_train).sum().item()
    accuracy = correct / len(y_train) * 100.0

    print(f'Epoch [{epoch+1}/{EPOCHS}], Loss: {loss.item():.4f}, Accuracy:
{accuracy:.2f}%')
```

```
  3%|▎         | 1/30 [00:12<05:59, 12.41s/it]
Epoch [1/30], Loss: 0.6984, Accuracy: 47.69%
  7%|▋         | 2/30 [00:33<08:17, 17.79s/it]
Epoch [2/30], Loss: 11.2522, Accuracy: 52.31%
 10%|█         | 3/30 [00:46<06:56, 15.43s/it]
Epoch [3/30], Loss: 2.9311, Accuracy: 52.22%
 13%|█▎        | 4/30 [00:58<06:09, 14.21s/it]
Epoch [4/30], Loss: 1.1631, Accuracy: 52.31%
```

```
 17%|█▋        | 5/30 [01:12<05:45, 13.81s/it]
Epoch [5/30], Loss: 2.0339, Accuracy: 48.20%
 20%|██        | 6/30 [01:24<05:20, 13.36s/it]
Epoch [6/30], Loss: 1.0975, Accuracy: 47.75%
 23%|██▎       | 7/30 [01:37<05:05, 13.29s/it]
Epoch [7/30], Loss: 0.6547, Accuracy: 48.29%
 27%|██▋       | 8/30 [01:49<04:45, 12.97s/it]
Epoch [8/30], Loss: 0.5906, Accuracy: 65.16%
 30%|███       | 9/30 [02:02<04:28, 12.80s/it]
Epoch [9/30], Loss: 0.5159, Accuracy: 85.44%
 33%|███▎      | 10/30 [02:15<04:18, 12.91s/it]
Epoch [10/30], Loss: 0.4104, Accuracy: 88.48%
 37%|███▋      | 11/30 [02:27<04:02, 12.76s/it]
Epoch [11/30], Loss: 0.2363, Accuracy: 94.74%
 40%|████      | 12/30 [02:41<03:53, 12.98s/it]
Epoch [12/30], Loss: 0.2008, Accuracy: 94.57%
 43%|████▎     | 13/30 [02:53<03:37, 12.81s/it]
Epoch [13/30], Loss: 0.1622, Accuracy: 95.46%
 47%|████▋     | 14/30 [03:06<03:23, 12.72s/it]
Epoch [14/30], Loss: 0.1197, Accuracy: 96.86%
 50%|█████     | 15/30 [03:19<03:14, 12.96s/it]
Epoch [15/30], Loss: 0.1070, Accuracy: 97.40%
 53%|█████▎    | 16/30 [03:32<02:59, 12.84s/it]
Epoch [16/30], Loss: 0.0793, Accuracy: 98.13%
 57%|█████▋    | 17/30 [03:45<02:48, 12.93s/it]
Epoch [17/30], Loss: 0.0550, Accuracy: 98.66%
 60%|██████    | 18/30 [03:58<02:34, 12.85s/it]
Epoch [18/30], Loss: 0.0440, Accuracy: 98.82%
 63%|██████▎   | 19/30 [04:10<02:20, 12.76s/it]
Epoch [19/30], Loss: 0.0373, Accuracy: 98.99%
 67%|██████▋   | 20/30 [04:24<02:09, 12.94s/it]
Epoch [20/30], Loss: 0.0289, Accuracy: 99.16%
 70%|███████   | 21/30 [04:36<01:55, 12.85s/it]
Epoch [21/30], Loss: 0.0243, Accuracy: 99.39%
 73%|███████▎  | 22/30 [04:49<01:43, 12.95s/it]
Epoch [22/30], Loss: 0.0211, Accuracy: 99.46%
 77%|███████▋  | 23/30 [05:02<01:29, 12.80s/it]
Epoch [23/30], Loss: 0.0153, Accuracy: 99.62%
 80%|████████  | 24/30 [05:14<01:16, 12.73s/it]
Epoch [24/30], Loss: 0.0105, Accuracy: 99.74%
 83%|████████▎ | 25/30 [05:28<01:04, 12.90s/it]
Epoch [25/30], Loss: 0.0073, Accuracy: 99.82%
 87%|████████▋ | 26/30 [05:40<00:51, 12.78s/it]
Epoch [26/30], Loss: 0.0055, Accuracy: 99.85%
 90%|█████████ | 27/30 [05:54<00:38, 12.99s/it]
```

```
Epoch [27/30], Loss: 0.0039, Accuracy: 99.89%
 93%|█████████▎| 28/30 [06:06<00:25, 12.86s/it]
Epoch [28/30], Loss: 0.0024, Accuracy: 99.94%
 97%|█████████▋| 29/30 [06:19<00:12, 12.83s/it]
Epoch [29/30], Loss: 0.0016, Accuracy: 99.96%
100%|██████████| 30/30 [06:33<00:00, 13.11s/it]
Epoch [30/30], Loss: 0.0011, Accuracy: 99.97%
```

Evaluating

In [27]:

```
model.eval()
with torch.no_grad():
    test_outputs = model(X_test)
    _, predicted = torch.max(test_outputs, 1)
    correct = (predicted == y_test).sum().item()
    test_accuracy = correct / len(y_test) * 100.0
    test_loss = criterion(test_outputs, y_test)

print(f'Test Accuracy: {test_accuracy:.2f}%')
print(f'Test Loss: {test_loss:.2f}%')
```

```
Test Accuracy: 99.21%
Test Loss: 0.04%
```

Deep learning modules Fake News Detection

Abstract:

- This module explores advanced techniques, specifically deep learning models, for improving the accuracy of fake news detection. It provides an abstract class, Advanced Fake News Detection, which defines methods for loading data, preprocessing data, training a model, making predictions, and evaluating the model's performance.

- Additionally, a concrete implementation of the abstract class called LSTM Fake News Detection is provided. This class extends Advanced Fake News Detection and utilizes an LSTM (Long Short-Term Memory) model for fake news detection.

- The LSTM Fake News Detection class loads data from a CSV file, preprocesses it by splitting it into train and test sets, vectorizes the text data using TF-IDF, and pads sequences for LSTM input. It then builds a Keras LSTM model, compiles it, and trains it on the training data.

- The predict method of the LSTM Fake News Detection class makes predictions using the trained model on the test data. The evaluate method calculates the accuracy of the model's predictions.

- To use this module, a CSV file containing fake news data needs to be provided, and the code needs to be modified accordingly.

PROGRAM & OUTPUT:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import tensorflow as tf
import re
from tensorflow.keras.preprocessing.text import
Tokenizer
import tensorflow as tf
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score,
confusion_matrix, precision_score, recall_score
import seaborn as sns
plt.style.use('ggplot')
```

Read the data

```
fake_df = pd.read_csvfake_df = pd.read_csv('fake-and-real-news-dataset/Fake.csv')
real_df = pd.read_csv('fake-and-real-news-dataset/True.csv')
```

```
fake_df.head(10)
```

	title	title	subject	Date
0	Donald Trump Sends Out Embarrassing New Year’...	Donald Trump just couldn t wish all Americans ...	News	December 31, 2017
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	December 30, 2017
3	Trump Is So Obsessed He Even Has Obama’s Name...	On Christmas day, Donald Trump announced that ...	News	December 29, 2017
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	December 25, 2017
5	Racist Alabama Cops Brutalize Black Boy While...	The number of cases of cops brutalizing and ki...	News	December 25, 2017

	title	title	subject	Date
6	Fresh Off The Golf Course, Trump Lashes Out A...	Donald Trump spent a good portion of his day a...	News	December 23, 2017
7	Trump Said Some INSANELY Racist Stuff Inside ...	In the wake of yet another court decision that...	News	December 23, 2017
8	Former CIA Director Slams Trump Over UN Bully...	Many people have raised the alarm regarding th...	News	December 22, 2017
9	WATCH: Brand-New Pro-Trump Ad Features So Muc...	Just when you might have thought we d get a br...	News	December 21, 2017

Fake News Detection

Importing Libraries

In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
import re
import string
```

Importing Dataset

```
In [2]:

df_fake = pd.read_csv("../input/fake-news-detection/Fake.csv")
df_true = pd.read_csv("../input/fake-news-detection/True.csv")
In [3]:

df_fake.head()
Out[3]:
```

	Title	Text	subject	Date
0	Donald Trump Sends Out Embarrassing New Year’...	Donald Trump just couldn t wish all Americans ...	News	December 31, 2017
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	December 30, 2017

	Title	Text	subject	Date
3	Trump Is So Obsessed He Even Has Obama’s Name...	On Christmas day, Donald Trump announced that ...	News	December 29, 2017
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	December 25, 2017

```
In [4]:
df_true.head(5)
Out[4]:
```

	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	WASHINGTON (Reuters) - Transgender people	politicsNews	December 29, 2017

	Title	text	subject	Date
	recruits o...	will...		
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

- Inserting a column "class" as target feature

```
In [5]:
df_fake["class"] = 0
```

```
df_true["class"] = 1
In [6]:
```

```
df_fake.shape, df_true.shape
Out[6]:
```

```
((23481, 5), (21417, 5))
In [7]:
```

```
# Removing last 10 rows for manual testing
df_fake_manual_testing = df_fake.tail(10)
for i in range(23480,23470,-1):
    df_fake.drop([i], axis = 0, inplace = True)


df_true_manual_testing = df_true.tail(10)
for i in range(21416,21406,-1):
    df_true.drop([i], axis = 0, inplace = True)
In [8]:
```

```
df_fake.shape, df_true.shape
Out[8]:
```

```
((23471, 5), (21407, 5))
In [9]:
```

```
df_fake_manual_testing["class"] = 0
df_true_manual_testing["class"] = 1


In [10]:
```

```
df_fake_manual_testing.head(10)
Out[10]:
```

	Title	text	subject	Date	class

	Title	text	subject	Date	class
23471	Seven Iranians freed in the prisoner swap have...	21st Century Wire says This week, the historic...	Middle-east	January 20, 2016	0
23472	#Hashtag Hell & The Fake Left	By Dady Chery and Gilbert MercierAll writers ...	Middle-east	January 19, 2016	0
23473	Astroturfing: Journalist Reveals Brainwashing ...	Vic Bishop Waking TimesOur reality is carefull...	Middle-east	January 19, 2016	0
23474	The New American Century: An Era of Fraud	Paul Craig RobertsIn the last years of the 20t...	Middle-east	January 19, 2016	0
23475	Hillary Clinton: ‘Israel First’ (and no peace	Robert Fantina Counterpunch Although the	Middle-east	January 18,	0

	Title	text	subject	Date	class
	...	United...		2016	
23476	McPain: John McCain Furious That Iran Treated ...	21st Century Wire says As 21WIRE reported earl...	Middle-east	January 16, 2016	0
23477	JUSTICE? Yahoo Settles E-mail Privacy Class-ac...	21st Century Wire says It s a familiar theme. ...	Middle-east	January 16, 2016	0
23478	Sunnistan: US and Allied ‘Safe Zone’ Plan to T...	Patrick Henningsen 21st Century WireRemember ...	Middle-east	January 15, 2016	0
23479	How to Blow $700 Million: Al Jazeera America F...	21st Century Wire says Al Jazeera America will...	Middle-east	January 14, 2016	0

	Title	text	subject	Date	class
23480	10 U.S. Navy Sailors Held by Iranian Military ...	21st Century Wire says As 21WIRE predicted in ...	Middle-east	January 12, 2016	0

```
In [11]:

df_true_manual_testing.head(10)
Out[11]:
```

	Title	text	subject	Date	class
21407	Mata Pires, owner of embattled Brazil builder ...	SAO PAULO (Reuters) - Cesar Mata Pires, the ow...	worldnews	August 22, 2017	1
21408	U.S., North Korea clash at U.N. forum over nuc...	GENEVA (Reuters) - North Korea and the United ...	worldnews	August 22, 2017	1

	Title	text	subject	Date	class
21409	U.S., North Korea clash at U.N. arms forum on ...	GENEVA (Reuters) - North Korea and the United ...	worldnews	August 22, 2017	1
21410	Headless torso could belong to submarine journ...	COPENHAGEN (Reuters) - Danish police said on T...	worldnews	August 22, 2017	1
21411	North Korea shipments to Syria chemical arms a...	UNITED NATIONS (Reuters) - Two North Korean sh...	worldnews	August 21, 2017	1
21412	'Fully committed' NATO backs new U.S. approach...	BRUSSELS (Reuters) - NATO allies on Tuesday we...	worldnews	August 22, 2017	1

	Title	text	subject	Date	class
21413	LexisNexis withdrew two products from Chinese ...	LONDON (Reuters) - LexisNexis, a provider of l...	worldnews	August 22, 2017	1
21414	Minsk cultural hub becomes haven from authorities	MINSK (Reuters) - In the shadow of disused Sov...	worldnews	August 22, 2017	1
21415	Vatican upbeat on possibility of Pope Francis ...	MOSCOW (Reuters) - Vatican Secretary of State ...	worldnews	August 22, 2017	1
21416	Indonesia to buy $1.14 billion worth of Russia...	JAKARTA (Reuters) - Indonesia will buy 11 Sukh...	worldnews	August 22, 2017	1

```
In [12]:
```

```
df_manual_testing =
pd.concat([df_fake_manual_testing,df_true_manual_test
ing], axis = 0)
df_manual_testing.to_csv("manual_testing.csv")
```

Merging True and Fake Dataframes

```
In [13]:
```

```
df_merge = pd.concat([df_fake, df_true], axis =0 )
df_merge.head(10)
Out[13]:
```

	Title	text	subject	Date	class
0	Donald Trump Sends Out Embarrassing New Year’...	Donald Trump just couldn t wish all Americans ...	News	December 31, 2017	0
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017	0
2	Sheriff David Clarke Becomes An	On Friday, it was revealed that former	News	December 30, 2017	0

	Title	text	subject	Date	class
	Internet Joke...	Milwauk...			
3	Trump Is So Obsessed He Even Has Obama’s Name...	On Christmas day, Donald Trump announced that ...	News	December 29, 2017	0
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	December 25, 2017	0
5	Racist Alabama Cops Brutalize Black Boy While...	The number of cases of cops brutalizing and ki...	News	December 25, 2017	0
6	Fresh Off The Golf Course, Trump Lashes Out A...	Donald Trump spent a good portion of his day a...	News	December 23, 2017	0

	Title	text	subject	Date	class
7	Trump Said Some INSANELY Racist Stuff Inside ...	In the wake of yet another court decision that...	News	December 23, 2017	0
8	Former CIA Director Slams Trump Over UN Bully...	Many people have raised the alarm regarding th...	News	December 22, 2017	0
9	WATCH: Brand-New Pro-Trump Ad Features So Muc...	Just when you might have thought we d get a br...	News	December 21, 2017	0

```
In [14]:

df_merge.columns
Out[14]:

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

- Removing columns which are not required

```
In [15]:

df = df_merge.drop(["title", "subject","date"], axis
= 1)
```

```
In [16]:
```

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

```
Out[16]:
```

```
text     0
class    0
dtype: int64
```

- Random Shuffling the dataframe

```
In [17]:
```

```
df = df.sample(frac = 1)
```

```
In [18]:
```

```
df.head()
```

```
Out[18]:
```

	Text	class
5099	During a live CNN interview with Rudy Giuliani...	0
1345	ANKARA (Reuters) - Turkey urged the United Sta...	1
20864	The attitudes of the family members defending ...	0
971	WASHINGTON (Reuters) - Charges brought against...	1

	Text	class
21217	The jurors in the Freddie Gray case were deadl...	0

```
In [19]:
```

```
df.reset_index(inplace = True)
df.drop(["index"], axis = 1, inplace = True)
```

```
In [20]:
```

```
df.columns
```

```
Out[20]:
```

```
Index(['text', 'class'], dtype='object')
```

```
In [21]:
```

```
df.head()
```

```
Out[21]:
```

	Text	class
0	During a live CNN interview with Rudy Giuliani...	0
1	ANKARA (Reuters) - Turkey urged the United Sta...	1
2	The attitudes of the family members defending ...	0

	Text	class
3	WASHINGTON (Reuters) - Charges brought against...	1
4	The jurors in the Freddie Gray case were deadl...	0

- Creating a function to process the texts

In [22]:

```
def wordopt(text):
    text = text.lower()
    text = re.sub('\[.*?\]', '', text)
    text = re.sub("\\W"," ",text)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s]' %
re.escape(string.punctuation), '', text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    return text
```

In [23]:

```
df["text"] = df["text"].apply(wordopt)
```

- Defining dependent and independent variables

In [24]:

```
x = df["text"]
y = df["class"]
```

- Splitting Training and Testing

```
In [25]:
```

```
x_train, x_test, y_train, y_test =
train_test_split(x, y, test_size=0.25)
```

- Convert text to vectors

```
In [26]:
```

```
from sklearn.feature_extraction.text import
TfidfVectorizer

vectorization = TfidfVectorizer()
xv_train = vectorization.fit_transform(x_train)
xv_test = vectorization.transform(x_test)
```

- Logistic Regression

```
In [27]:
```

```
from sklearn.linear_model import LogisticRegression

LR = LogisticRegression()
LR.fit(xv_train,y_train)
Out[27]:
```

```
LogisticRegression()
In [28]:
```

```
pred_lr=LR.predict(xv_test)
In [29]:
```

```
LR.score(xv_test, y_test)
Out[29]:
```

```
0.9885026737967915
In [30]:
```

```
print(classification_report(y_test, pred_lr))
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      5853
           1       0.99      0.99      0.99      5367

    accuracy                           0.99     11220
   macro avg       0.99      0.99      0.99     11220
weighted avg       0.99      0.99      0.99     11220
```

➢ Decision Tree Classification

```
In [31]:

from sklearn.tree import DecisionTreeClassifier

DT = DecisionTreeClassifier()
DT.fit(xv_train, y_train)
Out[31]:

DecisionTreeClassifier()
In [32]:

pred_dt = DT.predict(xv_test)
In [33]:

DT.score(xv_test, y_test)
Out[33]:

0.996524064171123
In [34]:

print(classification_report(y_test, pred_dt))
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      5853
           1       1.00      1.00      1.00      5367
```

```
    accuracy                           1.00     11220
   macro avg       1.00      1.00      1.00     11220
weighted avg       1.00      1.00      1.00     11220
```

- Gradient Boosting Classifier

In [35]:

```
from sklearn.ensemble import
GradientBoostingClassifier

GBC = GradientBoostingClassifier(random_state=0)
GBC.fit(xv_train, y_train)
```

Out[35]:

```
GradientBoostingClassifier(random_state=0)
```

In [36]:

```
pred_gbc = GBC.predict(xv_test)
```

In [37]:

```
GBC.score(xv_test, y_test)
```

Out[37]:

```
0.9959893048128342
```

In [38]:

```
print(classification_report(y_test, pred_gbc))
              precision    recall  f1-score   support

           0       1.00      0.99      1.00      5853
           1       0.99      1.00      1.00      5367

    accuracy                           1.00     11220
   macro avg       1.00      1.00      1.00     11220
weighted avg       1.00      1.00      1.00     11220
```

- Random Forest Classifier

```
In [39]:

from sklearn.ensemble import RandomForestClassifier

RFC = RandomForestClassifier(random_state=0)
RFC.fit(xv_train, y_train)
Out[39]:

RandomForestClassifier(random_state=0)
In [40]:

pred_rfc = RFC.predict(xv_test)
In [41]:

RFC.score(xv_test, y_test)
Out[41]:

0.9941176470588236
In [42]:

print(classification_report(y_test, pred_rfc))
              precision    recall  f1-score   support

           0       0.99      1.00      0.99      5853
           1       1.00      0.99      0.99      5367

    accuracy                           0.99     11220
   macro avg       0.99      0.99      0.99     11220
weighted avg       0.99      0.99      0.99     11220
```

- Model Testing

```
In [43]:

def output_lable(n):
    if n == 0:
        return "Fake News"
    elif n == 1:
        return "Not A Fake News"

def manual_testing(news):
    testing_news = {"text":[news]}
    new_def_test = pd.DataFrame(testing_news)
    new_def_test["text"] =
new_def_test["text"].apply(wordopt)
    new_x_test = new_def_test["text"]
    new_xv_test = vectorization.transform(new_x_test)
    pred_LR = LR.predict(new_xv_test)
    pred_DT = DT.predict(new_xv_test)
    pred_GBC = GBC.predict(new_xv_test)
    pred_RFC = RFC.predict(new_xv_test)

    return print("\n\nLR Prediction: {} \nDT
Prediction: {} \nGBC Prediction: {} \nRFC Prediction:
{}".format(output_lable(pred_LR[0]),
output_lable(pred_DT[0]),

output_lable(pred_GBC[0]),

output_lable(pred_RFC[0])))
In [44]:

news = str(input())
manual_testing(news)
BRUSSELS (Reuters) - NATO allies on Tuesday welcomed
President Donald Trump s decision to commit more
forces to Afghanistan, as part of a new U.S. strategy
he said would require more troops and funding from
America s partners. Having run for the White House
last year on a pledge to withdraw swiftly from
```

Afghanistan, Trump reversed course on Monday and promised a stepped-up military campaign against Taliban insurgents, saying: Our troops will fight to win . U.S. officials said he had signed off on plans to send about 4,000 more U.S. troops to add to the roughly 8,400 now deployed in Afghanistan. But his speech did not define benchmarks for successfully ending the war that began with the U.S.-led invasion of Afghanistan in 2001, and which he acknowledged had required an   extraordinary sacrifice of blood and treasure . We will ask our NATO allies and global partners to support our new strategy, with additional troops and funding increases in line with our own. We are confident they will, Trump said. That comment signaled he would further increase pressure on U.S. partners who have already been jolted by his repeated demands to step up their contributions to NATO and his description of the alliance as obsolete - even though, since taking office, he has said this is no longer the case. NATO Secretary General Jens Stoltenberg said in a statement: NATO remains fully committed to Afghanistan and I am looking forward to discussing the way ahead with (Defense) Secretary (James) Mattis and our Allies and international partners. NATO has 12,000 troops in Afghanistan, and 15 countries have pledged more, Stoltenberg said. Britain, a leading NATO member, called the U.S. commitment very welcome . In my call with Secretary Mattis yesterday we agreed that despite the challenges, we have to stay the course in Afghanistan to help build up its fragile democracy and reduce the terrorist threat to the West, Defence Secretary Michael Fallon said. Germany, which has borne the brunt of Trump s criticism over the scale of its

```
defense spending, also welcomed the new U.S. plan.
Our continued commitment is necessary on the path to
stabilizing the country,  a government spokeswoman
said. In June, European allies had already pledged
more troops but had not given details on numbers,
waiting for the Trump administration to outline its
strategy for the region.Nearly 16 years after the
U.S.-led invasion - a response to the Sept. 11
attacks which were planned by al Qaeda leader Osama
bin Laden from Afghanistan - the country is still
struggling with weak central government and a Taliban
insurgency. Trump said he shared the frustration of
the American people who were  weary of war without
victory , but a hasty withdrawal would create a
vacuum for groups like Islamic State and al Qaeda to
fill.


LR Prediction: Not A Fake News
DT Prediction: Not A Fake News
GBC Prediction: Not A Fake News
RFC Prediction: Not A Fake News
```

In [45]:

```python
news = str(input())
manual_testing(news)
```

```
Vic Bishop Waking TimesOur reality is carefully
constructed by powerful corporate, political and
special interest sources in order to covertly sway
public opinion. Blatant lies are often televised
regarding terrorism, food, war, health, etc. They are
fashioned to sway public opinion and condition
viewers to accept what have become destructive
societal norms.The practice of manipulating and
```

controlling public opinion with distorted media messages has become so common that there is a whole industry formed around this. The entire role of this brainwashing industry is to figure out how to spin information to journalists, similar to the lobbying of government. It is never really clear just how much truth the journalists receive because the news industry has become complacent. The messages that it presents are shaped by corporate powers who often spend millions on advertising with the six conglomerates that own 90% of the media:General Electric (GE), News-Corp, Disney, Viacom, Time Warner, and CBS. Yet, these corporations function under many different brands, such as FOX, ABC, CNN, Comcast, Wall Street Journal, etc, giving people the perception of choice   As Tavistock s researchers showed, it was important that the victims of mass brainwashing not be aware that their environment was being controlled; there should thus be a vast number of sources for information, whose messages could be varied slightly, so as to mask the sense of external control. ~ Specialist of mass brainwashing, L. WolfeNew Brainwashing Tactic Called AstroturfWith alternative media on the rise, the propaganda machine continues to expand. Below is a video of Sharyl Attkisson, investigative reporter with CBS, during which she explains how astroturf, or fake grassroots movements, are used to spin information not only to influence journalists but to sway public opinion. Astroturf is a perversion of grassroots. Astroturf is when political, corporate or other special interests disguise themselves and publish blogs, start facebook and twitter accounts, publish ads, letters to the editor, or simply post comments

```
online, to try to fool you into thinking an
independent or grassroots movement is speaking. ~
Sharyl Attkisson, Investigative ReporterHow do you
separate fact from fiction? Sharyl Attkisson finishes
her talk with some insights on how to identify signs
of propaganda and astroturfing  These methods are
used to give people the impression that there is
widespread support for an agenda, when, in reality,
one may not exist. Astroturf tactics are also used to
discredit or criticize those that disagree with
certain agendas, using stereotypical names such as
conspiracy theorist or quack. When in fact when
someone dares to reveal the truth or questions the
official  story, it should spark a deeper curiosity
and encourage further scrutiny of the
information.This article (Journalist Reveals Tactics
Brainwashing Industry Uses to Manipulate the Public)
was originally created and published by Waking Times
and is published here under a Creative Commons
license with attribution to Vic Bishop and
WakingTimes.com. It may be re-posted freely with
proper attribution, author bio, and this copyright
statement. READ MORE MSM PROPAGANDA NEWS AT: 21st
Century Wire MSM Watch Files


LR Prediction: Fake News
DT Prediction: Fake News
GBC Prediction: Fake News
RFC Prediction: Fake News
```

In [46]:

```python
news = str(input())
manual_testing(news)
```

SAO PAULO (Reuters) - Cesar Mata Pires, the owner and co-founder of Brazilian engineering conglomerate OAS SA, one of the largest companies involved in Brazil s corruption scandal, died on Tuesday. He was 68. Mata Pires died of a heart attack while taking a morning walk in an upscale district of S o Paulo, where OAS is based, a person with direct knowledge of the matter said. Efforts to contact his family were unsuccessful. OAS declined to comment. The son of a wealthy cattle rancher in the northeastern state of Bahia, Mata Pires links to politicians were central to the expansion of OAS, which became Brazil s No. 4 builder earlier this decade, people familiar with his career told Reuters last year. His big break came when he befriended Antonio Carlos Magalh es, a popular politician who was Bahia governor several times, and eventually married his daughter Tereza. Brazilians joked that OAS stood for Obras Arranjadas pelo Sogro - or Work Arranged by the Father-In-Law. After years of steady growth triggered by a flurry of massive government contracts, OAS was ensnared in Operation Car Wash which unearthed an illegal contracting ring between state firms and builders. The ensuing scandal helped topple former Brazilian President Dilma Rousseff last year. Trained as an engineer, Mata Pires founded OAS with two colleagues in 1976 to do sub-contracting work for larger rival Odebrecht SA - the biggest of the builders involved in the probe. Before the scandal, Forbes magazine estimated Mata Pires fortune at $1.6 billion. He dropped off the magazine s billionaire list in 2015, months after OAS sought bankruptcy protection after the Car Wash scandal. While Mata Pires was never accused of wrongdoing in the investigations,

```
creditors demanded he and his family stay away from
the builder s day-to-day operations, people directly
involved in the negotiations told Reuters at the
time. He is survived by his wife and his two sons.


LR Prediction: Not A Fake News
DT Prediction: Not A Fake News
GBC Prediction: Not A Fake News
RFC Prediction: Not A Fake News
```

Fake News Detection using LSTM based deep learning approach

➢ Abstract:

The identification of false information has become a critical concern in the modern era of technology, as the ready availability of information and widespread utilization of social media platforms have accelerated the dissemination of inaccurate news. The ability to accurately identify false news can help to mitigate the negative effects of misinformation, such as public confusion, political polarization, and potential harm to public health and safety. This paper presents a comprehensive review of ML and DL based approaches for fake news detection. Our review provides insights and guidance for researchers and practitioners interested in developing effective fake news detection systems using ML and DL approaches. News reporters often need to verify authenticity of news stories before publishing or reporting them. By utilizing fake news detection models, reporters can filter out fake news and focus on reporting accurate and reliable information.

➢ Introduction:

This research investigates the application of ML and DL algorithms in detecting fake news. The study initially explores the characteristics of fake news and the challenges of detecting it. Then, it presents an overview of ML algorithms and their application to fake news detection. The study evaluates performance of several algorithms on a dataset of real and fake news articles. Results specify that ML algorithms cannot effectively distinguish between real and fake news articles with the high accuracy. As such, we put forth a deep learning methodology that uses LSTM neural networks for identifying false news. The proposed approach takes the textual content of news articles as input and utilizes an LSTM architecture to capture the temporal dependencies of the text. The proposed LSTM-based model is trained on a dataset of news articles from Kaggle and achieved an accuracy of 94% in detecting fake news. This performance is a significant improvement over previous approaches for fake news detection.

The proposed approach is beneficial in real world scenarios where there is a high volume of news articles to analyse. It can also be useful for social media platforms to detect and remove fake news from their networks. Model's ability to capture the temporal dependencies of the text is especially relevant in the context of news articles, where the order of the words and phrases can significantly impact the article's meaning. Our study contributes to the ongoing efforts in combatting the spread of misinformation and highlights the potential of DL approaches in detecting fake news. Proposed LSTM-based model is a promising tool for identifying fake news, and it can also be extended to other domains, such as social media posts and online reviews, where fake or malicious content can also spread. By deploying the proposed model in real-world applications, we can help users make informed decisions and reduce the impact of fake news. Overall, this study highlights potential of DL approaches in detecting false news and contributes to the ongoing efforts in combatting the spread of misinformation in digital media.

➢ Methodology:

The methodology is commonly utilized for the purpose of identifying fake news, as it consists of a compilation of news articles accompanied by labels that indicate whether the news is genuine or fraudulent. With this data, machine learning models can be trained to recognize patterns in the text and make predictions about the authenticity of news articles. The dataset has been compiled from several reliable sources such as Politico, NPR, CNN, and Reuters. It is well-curated, and the sources are trusted news outlets. The dataset contains a mix of news articles from different categories, including politics, business, and entertainment, among others

- **Import necessary libraries**:

The code begins by importing the required libraries, including pandas for data manipulation, seaborn and matplotlib for visualization of data, and various modules from nltk for text preprocessing.

- **Load and preprocess the data:**

The news data is loaded from a CSV file using the pandas library. The 'title' column is dropped from the data, and the remaining columns are checked for any missing values. The 'text' column is then preprocessed using the preprocess_text() function, which performs text cleaning, tokenization, and stopword removal.

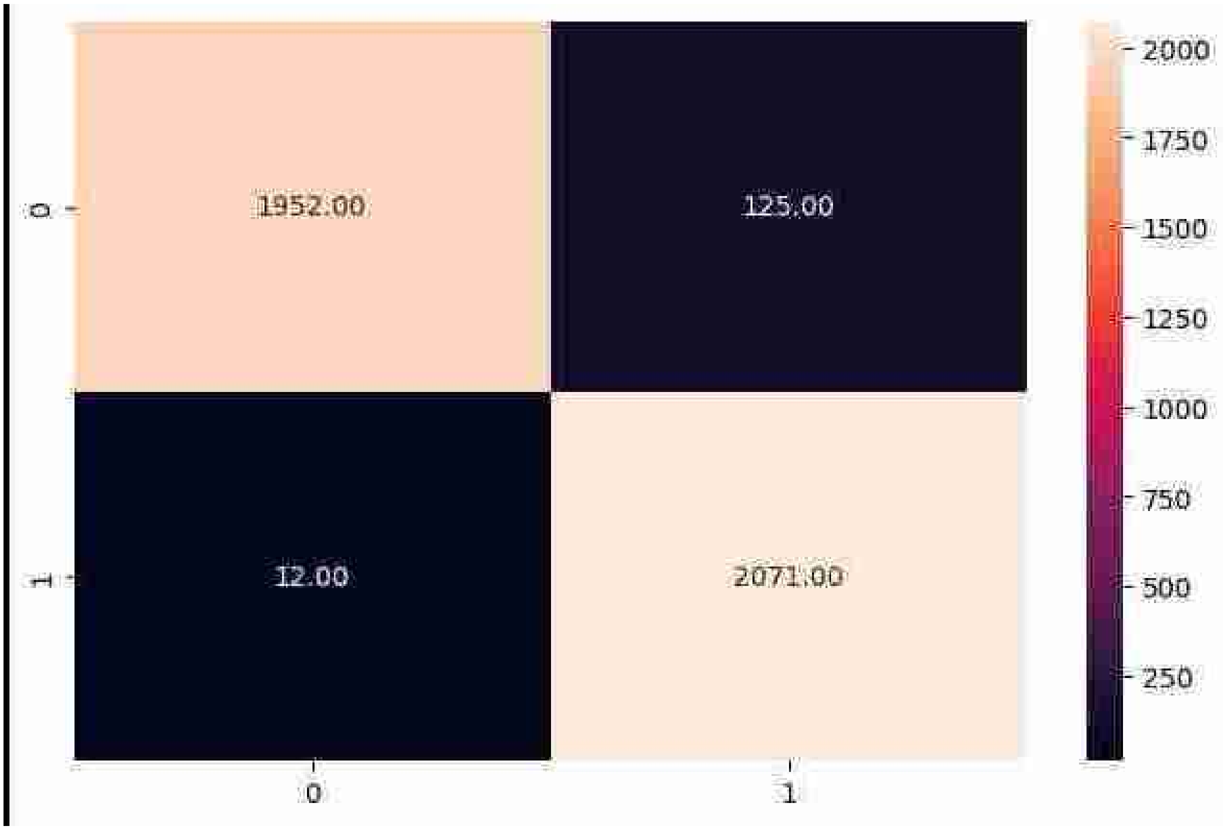
- **Generate word clouds**:

Word clouds are generated for both the 'REAL' and 'FAKE' labels using the Word Cloud library. Word clouds are visual representations of the most frequently occurring words in a text, with word size indicating word frequency. These word clouds provide a visual summary of most common words in the news articles for each label.

- **Analyze stop words:**

The most common words from the news articles are analyzed using get_top_n_words() function, which uses CountVectorizer to convert text data into a bag of words representation and calculates the word frequencies. A bar chart is then plotted to display the top words and their frequencies.

- **Train and evaluate ML models**: Sklearn.model_selection's train_test_split() is used to partition the data into training and testing sets.. The text data is vectorized using TfidfVectorizer, and two ML models, Logistic Regression and Decision Tree Classifier, are trained and evaluated using accuracy_score. The Confusion Matrix is also plotted to visualize performance of Decision Tree Classifier



We noticed, Machine learning algorithms can use previous data during training to learn patterns, but they do not inherently have built-in memory to explicitly store and recall previous data points during prediction, which is a capability that deep learning models, specifically designed for sequential data. So, we implemented the LSTM model in order to increase the performance of fake news detection system. LSTM models can capture long-term dependencies and patterns in sequential data, making them suitable for text classification tasks like news classification. LSTM models can be implemented using deep learning libraries such as Keras or PyTorch. The code trains an LSTM model on a dataset consisting of news articles that are labeled to indicate their authenticity (real or fake). The code follows following series of steps:

Load and preprocess the dataset: The news dataset is loaded from a CSV file and only the first 1000 rows are used. The null values are dropped and the index is reset. The label column is converted to binary values, 1 represents real news and 0 represents false news.

Divide the data into training dataset and testing dataset: The data is allocated into 80% for training and 20% for testing

Tokenize and pad the text data:

Tokenization is the process of converting text into numerical data which the neural network can process. The text is tokenized using Keras Tokenizer class and then padded to ensure that all sequences have the same length.

Define the LSTM model architecture:

To define the model, a Keras Sequential model is utilized. The initial layer is an embedding layer that maps the tokenized text into a dense vector. Following this, a bidirectional LSTM layer is implemented to capture the contextual information of the text. To increase the non-linearity of the model, a dense layer with ReLU activation function is included. Finally, a dense layer with sigmoid activation function is added to produce a binary output that indicates whether the news is real or fake. • Deciding the value of epoch: We started by defining a maximum number of epochs that we wanted to allow for training. • Update epoch: We checked if the current accuracy meets the desired threshold. If it did, we finalized the current epoch value.

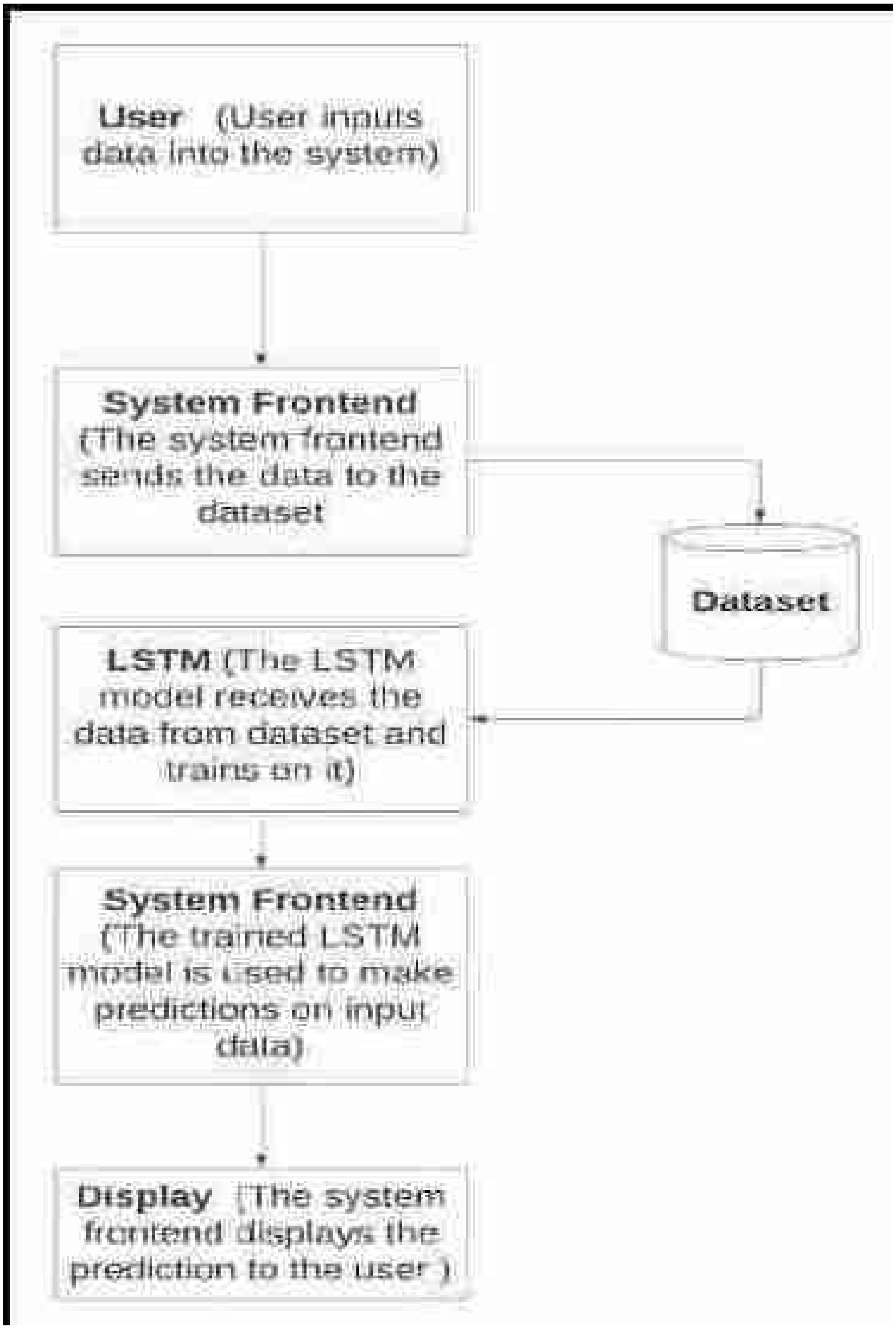
Compile the model: Binary cross-entropy loss function along with Adam optimizer and accuracy metric are used to compile the model.

Train the model: The training set is used to train the model for a specified number of epochs.

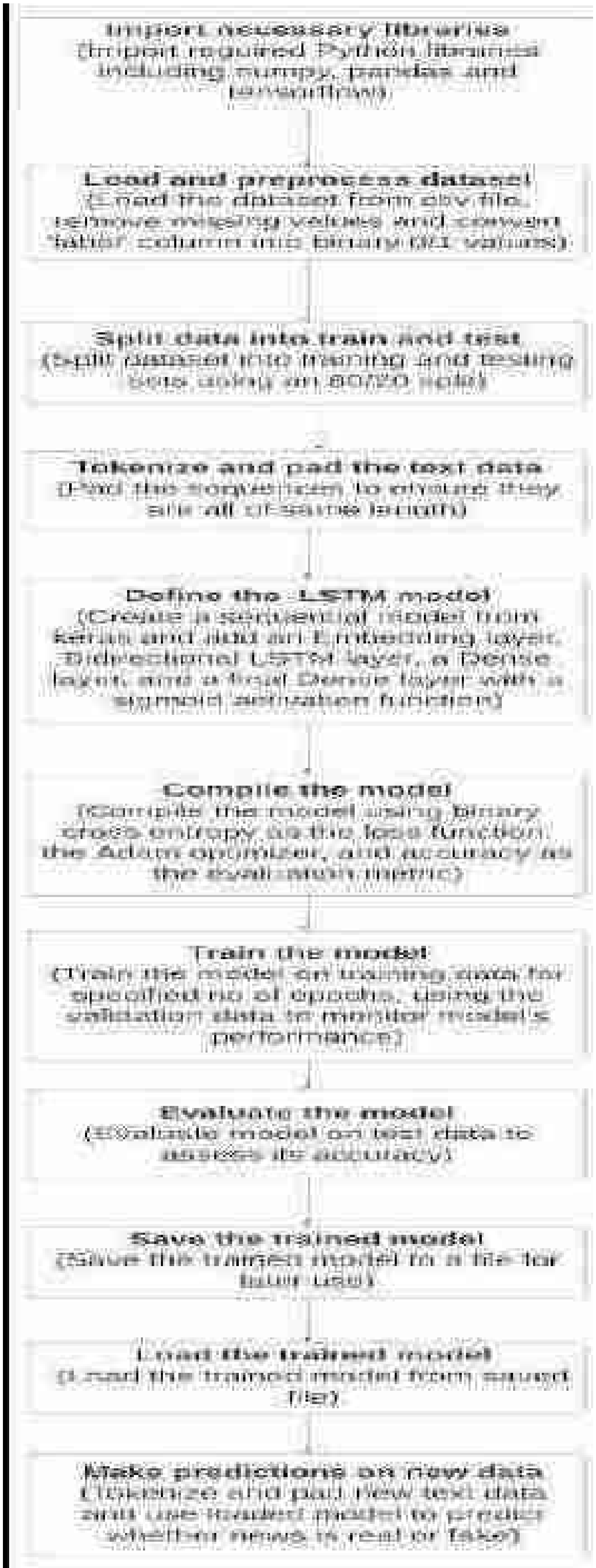
Evaluate the model: The model's loss and accuracy of predictions are assessed to evaluate its performance.

. **Save the trained model:** The trained model is saved in a file with a .h5 extension.
Load the trained model: The saved model is loaded back into memory.
Make predictions on new data: The loaded model is used to make predictions on new data, such as text. Text is tokenized and padded before being passed to the model. The output is a binary value indicating whether the result is true or false.

Flowcharts and chart of model:


User (User inputs data into the system)
System Frontend (The system frontend sends the data to the dataset
Dataset
LSTM (The LSTM model receives the data from dataset and trains on it)
System Frontend (The trained LSTM model is used to make predictions on input data)
Display (The system frontend displays the prediction to the user)

Architecture of System :


Import necessary libraries
(Import required Python libraries including numpy, pandas and tensorflow)
Load and preprocess dataset
(Load the dataset from csv file, remove missing values and convert 'label' column into binary 0/1 values)
Split data into train and test
(Split dataset into training and testing sets using an 80/20 split)
Tokenize and pad the text data
(Pad the sequences to ensure they are all of same length)
Define the LSTM model
(Create a sequential model from keras and add an Embedding layer, Bidirectional LSTM layer, a Dense layer, and a final Dense layer with a sigmoid activation function)
Compile the model
(Compile the model using binary cross entropy as the loss function, the Adam optimizer, and accuracy as the evaluation metric)
Train the model
(Train the model on training data for specified no of epochs, using the validation data to monitor model's performance)
Evaluate the model
(Evaluate model on test data to assess its accuracy)
Save the trained model
(Save the trained model to a file for later use)
Load the trained model
(Load the trained model from saved file)
Make predictions on new data
(Tokenize and pad new text data and use loaded model to predict whether news is real or fake)

- **Results:** The study aimed to develop a fake news detection system using various machine learning (ML) algorithms and a deep learning model, Long Short-Term Memory (LSTM). The study utilized a dataset consisting of news articles labeled as real or fake to train and evaluate the models. The following ML algorithms were Regression, Random Forest, Multinomial Naive Bayes, and K-Nearest Neighbors Classifier. The models were evaluated based on their training and testing accuracy, precision, recall, F1-score, true negative rate, and false positive rate. The LSTM model was also implemented and evaluated. The Decision Tree and Random Forest models showed 100% training accuracy, indicating that they overfit the data. The Logistic Regression model showed the highest testing accuracy of 0.9122, followed by the Random Forest model with 0.8907. The Multinomial Naive Bayes and K-Nearest Neighbors Classifier models showed lower testing accuracies of 0.7941 and 0.7468, respectively. These results suggest that Logistic Regression and Random Forest are better suited for fake news detection than the other ML algorithms tested in this study. Figure 1 illustrates the confusion matrix of the Decision Tree Classifier. It provides a visual representation of the model's true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) counts. The model correctly identified 659 real news articles and 628 fake news articles. However, it misclassified 90 real news articles as fake news and 103 fake news articles as real news. The LSTM model was trained on the same dataset as the ML algorithms. The dataset was preprocessed by tokenizing and padding the text data. The LSTM model showed an accuracy of 0.8667 on the testing set. The LSTM model outperformed the Multinomial Naive Bayes and K-Nearest Neighbors Classifier models in terms of testing accuracy. The results suggest that the LSTM model has potential for use in fake news detection.

Load and Preprocess Dataset for Fake News Detection

Introduction:

- Provide a brief overview of the project, including the goals and objectives.
- Discuss the importance of fake news detection and the role that machine learning can play in this task.
- Introduce the dataset that will be used in the project.

Data Loading:

- Describe how the dataset was loaded into the Python environment.
- Explain any data cleaning steps that were performed.

Text Preprocessing:

- Describe how the text in each news article was preprocessed.
- Explain the rationale for each preprocessing step.

Vectorization:

- Describe how the text in each news article was converted to a vector of word frequencies.
- Explain the rationale for using the bag-of-words approach.
- Splitting the Dataset into Training and Testing Sets
- Describe how the dataset was split into training and testing sets.
- Explain the importance of using a separate testing set to evaluate the model.

PROGRAM &OUTPUT:

```
import  numpy as np
import pandas as pd
import  itertools
from sklearn.model_selection import train_test_split
from sklearn.linear_model import PassiveAggressiveClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
import string as st
import re
import nltk
from nltk import PorterStemmer, WordNetLemmatizer
import matplotlib.pyplot as plt

import os
```

```
for dirname, _, filenames in os.walk('input of fake data set'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import itertools
from sklearn.model_selection import train_test_split
from sklearn.linear_model import PassiveAggressiveClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
import string as st
import re
import nltk
from nltk import PorterStemmer, WordNetLemmatizer
import matplotlib.pyplot as plt


import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
In [2]:
data = pd.read_csv('../input/textdb3/fake_or_real_news.csv')
data.shape
```

```
Out[2]:
(6335, 4)
```

```
In [3]:
data.head()
```

```
Out[3]:
```

	Unnamed: 0	title	text	label
0	8476	You Can Smell Hillary’s Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE
1	10294	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg Linkedin Reddit Stumbleu...	FAKE
2	3608	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	REAL
3	10142	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE

	Unnamed: 0	title	text	label
4	875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL

```
In [4]:
# Check how the labels are distributed
print(np.unique(data['label']))
print(np.unique(data['label'].value_counts()))
```

```
['FAKE' 'REAL']
[3164 3171]
```

Text cleaning and processing steps-

- Remove punctuations
- Convert text to tokens
- Remove tokens of length less than or equal to 3
- Remove stopwords using NLTK corpus stopwords list to match
- Apply lemmatization
- Convert words to feature vectors

```
In [5]:
# Remove all punctuations from the text

def remove_punct(text):
    return ("".join([ch for ch in text if ch not in st.punctuation]))
```

```
In [6]:
data['removed_punc'] = data['text'].apply(lambda x: remove_punct(x))
data.head()
```

```
Out[6]:
```

	Unnamed: 0	title	text	label	removed_punc
0	8476	You Can Smell Hillary’s Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE	Daniel Greenfield a Shillman Journalism Fellow...
1	10294	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg Linkedin Reddit Stumbleu...	FAKE	Google Pinterest Digg Linkedin Reddit Stumbleu...

	Unnamed: 0	title	text	label	removed_punc
2	3608	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	REAL	US Secretary of State John F Kerry said Monday...
3	10142	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE	— Kaydee King KaydeeKing November 9 2016 The l...
4	875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL	Its primary day in New York and frontrunners H...

In [7]:

```
''' Convert text to lower case tokens. Here, split() is applied on white-spaces. But,
it could be applied
    on special characters, tabs or any other string based on which text is to be
seperated into tokens.
'''
def tokenize(text):
    text = re.split('\s+' ,text)
    return [x.lower() for x in text]
```

In [8]:

```
data['tokens'] = data['removed_punc'].apply(lambda msg : tokenize(msg))
data.head()
```

Out[8]:

	Unnamed: 0	title	text	label	removed_punc	tokens
0	8476	You Can Smell Hillary’s Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE	Daniel Greenfield a Shillman Journalism Fellow...	[daniel, greenfield, a, shillman, journalism, ...
1	10294	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg Linkedin Reddit Stumbleu...	FAKE	Google Pinterest Digg Linkedin Reddit Stumbleu...	[google, pinterest, digg, linkedin, reddit, st...

	Unnamed: 0	title	text	label	removed_punc	tokens
2	3608	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	REAL	US Secretary of State John F Kerry said Monday...	[us, secretary, of, state, john, f, kerry, sai...
3	10142	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE	— Kaydee King KaydeeKing November 9 2016 The l...	[—, kaydee, king, kaydeeking, november, 9, 201...
4	875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL	Its primary day in New York and frontrunners H...	[its, primary, day, in, new, york, and, frontr...

```
In [9]:
# Remove tokens of length less than 3
def remove_small_words(text):
    return [x for x in text if len(x) > 3 ]
```

```
In [10]:
data['filtered_tokens'] = data['tokens'].apply(lambda x : remove_small_words(x))
data.head()
```

```
Out[10]:
```

	Unnamed: 0	title	text	label	removed_punc	tokens	filtered_tokens
0	8476	You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE	Daniel Greenfield a Shillman Journalism Fellow...	[daniel, greenfield, a, shillman, journalism, ...	[daniel, greenfield, shillman, journalism, fel...
1	10294	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg Linkedin Reddit Stumbleu...	FAKE	Google Pinterest Digg Linkedin Reddit Stumbleu...	[google, pinterest, digg, linkedin, reddit, st...	[google, pinterest, digg, linkedin, reddit, st...
2	3608	Kerry to go to Paris in	U.S. Secretary of State John F. Kerry	REAL	US Secretary of State John F	[us, secretary, of, state, john,	[secretary, state, john, kerry, said,

	Unnamed: 0	title	text	label	removed_punc	tokens	filtered_tokens
		gesture of sympathy	said Mon...		Kerry said Monday...	f, kerry, sai...	monday, ...
3	10142	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE	— Kaydee King KaydeeKing November 9 2016 The l...	[—, kaydee, king, kaydeeking, november, 9, 201...	[kaydee, king, kaydeeking, november, 2016, les...
4	875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL	Its primary day in New York and frontrunners H...	[its, primary, day, in, new, york, and, frontr...	[primary, york, frontrunners, hillary, clinton...

```
In [11]:
''' Remove stopwords. Here, NLTK corpus list is used for a match. However, a
customized user-defined
    list could be created and used to limit the matches in input text.
'''
def remove_stopwords(text):
    return [word for word in text if word not in
nltk.corpus.stopwords.words('english')]
```

```
In [12]:
data['clean_tokens'] = data['filtered_tokens'].apply(lambda x : remove_stopwords(x))
data.head()
```

Out[12]:

	Unnamed: 0	title	text	label	removed_punc	tokens	filtered_tokens	clean_tokens
0	8476	You Can Smell Hillary’s Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE	Daniel Greenfield a Shillman Journalism Fellow...	[daniel, greenfield, a, shillman, journalism, ...	[daniel, greenfield, shillman, journalism, fel...	[daniel, greenfield, shillman, journalism, fel...

	Unnamed: 0	title	text	label	removed_punc	tokens	filtered_tokens	clean_tokens
1	10294	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg Linkedin Reddit Stumbleu...	FAKE	Google Pinterest Digg Linkedin Reddit Stumbleu...	[google, pinterest, digg, linkedin, reddit, st...	[google, pinterest, digg, linkedin, reddit, st...	[google, pinterest, digg, linkedin, reddit, st...
2	3608	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	REAL	US Secretary of State John F Kerry said Monday...	[us, secretary, of, state, john, f, kerry, sai...	[secretary, state, john, kerry, said, monday, ...	[secretary, state, john, kerry, said, monday, ...
3	10142	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE	— Kaydee King KaydeeKing November 9 2016 The l...	[—, kaydee, king, kaydeeking, november, 9, 201...	[kaydee, king, kaydeeking, november, 2016, les...	[kaydee, king, kaydeeking, november, 2016, les...
4	875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL	Its primary day in New York and frontrunners H...	[its, primary, day, in, new, york, and, frontr...	[primary, york, frontrunners, hillary, clinton...	[primary, york, frontrunners, hillary, clinton...

In [13]:

```
# Apply lemmatization on tokens
def lemmatize(text):
    word_net = WordNetLemmatizer()
    return [word_net.lemmatize(word) for word in text]
```

In [14]:

```
data['lemma_words'] = data['clean_tokens'].apply(lambda x : lemmatize(x))
data.head()
```

Out[14]:

	Unnamed: 0	title	text	label	removed_punc	tokens	filtered_tokens	clean_tokens	lemma_words

	Unnamed: 0	title	text	label	removed_punc	tokens	filtered_tokens	clean_tokens	lemma_words
0	8476	You Can Smell Hillary’s Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE	Daniel Greenfield a Shillman Journalism Fellow...	[daniel, greenfield, a, shillman, journalism, ...	[daniel, greenfield, shillman, journalism, fel...	[daniel, greenfield, shillman, journalism, fel...	[daniel, greenfield, shillman, journalism, fel...
1	10294	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg Linkedin Reddit Stumbleu...	FAKE	Google Pinterest Digg Linkedin Reddit Stumbleu...	[google, pinterest, digg, linkedin, reddit, st...	[google, pinterest, digg, linkedin, reddit, st...	[google, pinterest, digg, linkedin, reddit, st...	[google, pinterest, digg, linkedin, reddit, st...
2	3608	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	REAL	US Secretary of State John F Kerry said Monday...	[us, secretary, of, state, john, f, kerry, sai...	[secretary, state, john, kerry, said, monday, ...	[secretary, state, john, kerry, said, monday, ...	[secretary, state, john, kerry, said, monday, ...
3	10142	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE	— Kaydee King KaydeeKing November 9 2016 The l...	[—, kaydee, king, kaydeeking, november, 9, 201...	[kaydee, king, kaydeeking, november, 2016, les...	[kaydee, king, kaydeeking, november, 2016, les...	[kaydee, king, kaydeeking, november, 2016, les...
4	875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL	Its primary day in New York and frontrunners H...	[its, primary, day, in, new, york, and, frontr...	[primary, york, frontrunners, hillary, clinton...	[primary, york, frontrunners, hillary, clinton...	[primary, york, frontrunners, hillary, clinton...

In [15]:

```
# Create sentences to get clean text as input for vectors
```

```
def return_sentences(tokens):
    return " ".join([word for word in tokens])
```

In [16]:

```
data['clean_text'] = data['lemma_words'].apply(lambda x : return_sentences(x))
data.head()
```

```
import pandas as pd
import numpy as np
import re
import nltk
df = pd.read_csv('fake_news_dataset.csv')
df.head()
df.shape
df.columns
df.dtypes
df['news'] = df['news'].apply(lambda x: re.sub(r'[^\w\s]', '', x))
df['news'] = df['news'].apply(lambda x: x.lower())
stopwords = nltk.corpus.stopwords.words('english')
df['news'] = df['news'].apply(lambda x: ' '.join([word for word in x.split() if
word not in stopwords]))
df['news'] = df['news'].apply(lambda x: ' '.join([word for word in x.split() if
len(word) > 2]))
lemmatizer = nltk.WordNetLemmatizer()
df['news'] = df['news'].apply(lambda x: ' '.join([lemmatizer.lemmatize(word) for
word in x.split()]))
df.to_csv('fake_news_preprocessed.csv', index=False)
```

Out[16]:

	Unnamed: 0	title	text	label	removed_punc	tokens	filtered_tokens	clean_tokens	lemma_words	clean_text
0	8476	You Can Smell Hillary’s Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE	Daniel Greenfield a Shillman Journalism Fellow...	[daniel, greenfield, a, shillman, journalism, ...	[daniel, greenfield, shillman, journalism, fel...	[daniel, greenfield, shillman, journalism, fel...	[daniel, greenfield, shillman, journalism, fel...	daniel greenfield shillman journalism fellow f...
1	10294	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg Linkedin Reddit Stumbleu...	FAKE	Google Pinterest Digg Linkedin Reddit Stumbleu...	[google, pinterest, digg, linkedin, reddit, st...	[google, pinterest, digg, linkedin, reddit, st...	[google, pinterest, digg, linkedin, reddit, st...	[google, pinterest, digg, linkedin, reddit, st...	google pinterest digg linkedin reddit stumbleu...
2	3608	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	REAL	US Secretary of State John F Kerry said Monday...	[us, secretary, of, state, john, f, kerry, sai...	[secretary, state, john, kerry, said, monday, ...	[secretary, state, john, kerry, said, monday, ...	[secretary, state, john, kerry, said, monday, ...	secretary state john kerry said monday stop pa...
3	10142	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE	— Kaydee King KaydeeKing November 9 2016 The l...	[—, kaydee, king, kaydeeking, november, 9, 201...	[kaydee, king, kaydeeking, november, 2016, les...	[kaydee, king, kaydeeking, november, 2016, les...	[kaydee, king, kaydeeking, november, 2016, les...	kaydee king kaydeeking november 2016 lesson to...
4	875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL	Its primary day in New York and frontrunners H...	[its, primary, day, in, new, york, and, frontr...	[primary, york, frontrunners, hillary, clinton...	[primary, york, frontrunners, hillary, clinton...	[primary, york, frontrunners, hillary, clinton...	primary york frontrunners hillary clinton dona...

```
In [17]:
from wordcloud import WordCloud, ImageColorGenerator

text = " ".join([x for x in data['clean_text']])
wordcloud = WordCloud(max_font_size=30, max_words=1000).generate(text)

plt.figure(figsize= [20,10])
plt.imshow(wordcloud)
plt.axis("off")
plt.show()
```

```
In [18]:

data['label'] = [1 if x == 'FAKE' else 0 for x in data['label']]
data.head()
```

```
Out[18]:
```

	Unnamed: 0	title	text	label	removed_punc	tokens	filtered_tokens	clean_tokens	lemma_words	clean_text
0	8476	You Can Smell Hillary’s Fear	Daniel Greenfield, a Shillman Journalism Fello...	1	Daniel Greenfield a Shillman Journalism Fellow...	[daniel, greenfield, a, shillman, journalism, ...	[daniel, greenfield, shillman, journalism, fel...	[daniel, greenfield, shillman, journalism, fel...	[daniel, greenfield, shillman, journalism, fel...	daniel greenfield shillman journalism fellow f...
1	10294	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg Linkedin Reddit Stumbleu...	1	Google Pinterest Digg Linkedin Reddit Stumbleu...	[google, pinterest, digg, linkedin, reddit, st...	[google, pinterest, digg, linkedin, reddit, st...	[google, pinterest, digg, linkedin, reddit, st...	[google, pinterest, digg, linkedin, reddit, st...	google pinterest digg linkedin reddit stumbleu ...
2	3608	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	0	US Secretary of State John F Kerry said Monday...	[us, secretary, of, state, john, f, kerry, sai...	[secretary, state, john, kerry, said, monday, ...	[secretary, state, john, kerry, said, monday, ...	[secretary, state, john, kerry, said, monday, ...	secretary state john kerry said monday stop pa...

	Unnamed: 0	title	text	label	removed_punc	tokens	filtered_tokens	clean_tokens	lemma_words	clean_text
3	10142	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	1	— Kaydee King KaydeeKing November 9 2016 The l...	[—, kaydee, king, kaydeeking, november, 9, 201...	[kaydee, king, kaydeeking, november, 2016, les...	[kaydee, king, kaydeeking, november, 2016, les...	[kaydee, king, kaydeeking, november, 2016, les...	kaydee king kaydeeking november 2016 lesson to...
4	875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	0	Its primary day in New York and frontrunners H...	[its, primary, day, in, new, york, and, frontr...	[primary, york, frontrunners, hillary, clinton...	[primary, york, frontrunners, hillary, clinton...	[primary, york, frontrunners, hillary, clinton...	primary york frontrunners hillary clinton dona...

```
In [19]:

X_train,X_test,y_train,y_test = train_test_split(data['clean_text'], data['label'],
test_size=0.2, random_state = 5)

print(X_train.shape)
print(X_test.shape)
```

```
(5068,)
(1267,)
```

TF-IDF : Term Frequency - Inverse Document Frequency

The term frequency is the number of times a term occurs in a document. Inverse document frequency is an inverse function of the number of documents in which that a given word occurs.

The product of these two terms gives tf-idf weight for a word in the corpus. The higher the frequency of occurrence of a word, lower is it's weight and vice-versa. This gives more weightage to rare terms in the corpus and penalizes more commonly occuring terms.

Other widely used vectorizer is Count vectorizer which only considers the frequency of occurrence of a word across the corpus.

```
In [20]:
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf = TfidfVectorizer()
```

```
tfidf_train = tfidf.fit_transform(X_train)
tfidf_test = tfidf.transform(X_test)

print(tfidf_train.toarray())
print(tfidf_train.shape)
print(tfidf_test.toarray())
print(tfidf_test.shape)
```

```
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
(5068, 68134)
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
(1267, 68134)
```

Passive Aggressive Classifiers

- Passive Aggressive algorithms are online learning algorithms. Such an algorithm remains passive for a correct classification outcome, and turns aggressive in the event of a miscalculation, updating and adjusting.
- Passive: If the prediction is correct, keep the model and do not make any changes. i.e., the data in the example is not enough to cause any changes in the model.
- Aggressive: If the prediction is incorrect, make changes to the model. i.e., some change to the model may correct it.

These are typically used for large datasets where batch learning is not possible due to huge volumes of frequently incoming data.

- Some important parameters -
- C : This is the regularization parameter, and denotes the penalization the model will make on an incorrect prediction
- max_iter : The maximum number of iterations the model makes over the training data.
- tol : The stopping criterion.

```
In [21]:
# Passive Aggresive Classifier
pac = PassiveAggressiveClassifier(max_iter=50)
pac.fit(tfidf_train,y_train)

pred = pac.predict(tfidf_test)
print("Accuracy score : {}".format(accuracy_score(y_test, pred)))
print("Confusion matrix : \n {}".format(confusion_matrix(y_test, pred)))
```

```
Accuracy score : 0.936069455406472
Confusion matrix :
 [[592  38]
 [ 43 594]]
```

In [22]:

```python
# Logistic Regression model
from sklearn.linear_model import LogisticRegression

lr = LogisticRegression(max_iter = 500)
lr.fit(tfidf_train, y_train)
print('Logistic Regression model fitted..')

pred = lr.predict(tfidf_test)
print("Accuracy score : {}".format(accuracy_score(y_test, pred)))
print("Confusion matrix : \n {}".format(confusion_matrix(y_test, pred)))
```

```
Logistic Regression model fitted..
Accuracy score : 0.9179163378058406
Confusion matrix :
 [[565  65]
 [ 39 598]]
```

Logistic Regression could not outperform XGBoost and LGBM but its performance is considerably close to them and it is much less complex.

In [23]:

```python
import xgboost
from xgboost import XGBClassifier

xgb = XGBClassifier()
xgb.fit(tfidf_train, y_train)

print('XGBoost Classifier model fitted..')
pred = xgb.predict(tfidf_test)
print("Accuracy score : {}".format(accuracy_score(y_test, pred)))
print("Confusion matrix : \n {}".format(confusion_matrix(y_test, pred)))
```

```
XGBoost Classifier model fitted..
Accuracy score : 0.9289660615627466
Confusion matrix :
 [[587  43]
 [ 47 590]]
```

In [24]:

```python
import lightgbm
from lightgbm import LGBMClassifier

lgbm = LGBMClassifier()
lgbm.fit(tfidf_train, y_train)

print('LightGBM Classifier model fitted..')
pred = lgbm.predict(tfidf_test)
print("Accuracy score : {}".format(accuracy_score(y_test, pred)))
print("Confusion matrix : \n {}".format(confusion_matrix(y_test, pred)))
```

```
LightGBM Classifier model fitted..
Accuracy score : 0.9289660615627466
```

```
Confusion matrix :
 [[581  49]
 [ 41 596]]
```

Fake News Detection using NLP and Machine Learning

Introduction:

Fake news is a type of misinformation that is intentionally or unintentionally spread through various media platforms, such as social networks, blogs, websites, etc. Fake news can have negative impacts on society, such as influencing public opinion, undermining trust in institutions, and inciting violence. Therefore, it is important to develop methods to detect and prevent the spread of fake news.

One of the challenges in fake news detection is that fake news can be written in various styles, such as satire, propaganda, click bait, etc. Moreover, fake news can be mixed with some factual information to make it more convincing. Therefore, traditional methods based on keywords or rules may not be effective in identifying fake news.

In this project, we propose to use natural language processing (NLP) and machine learning techniques to build a fake news detection model. NLP is a branch of artificial intelligence that deals with the analysis and generation of natural language. Machine learning is a branch of artificial intelligence that enables computers to learn from data and make predictions. By combining NLP and machine learning, we aim to extract useful features from the text of news articles and train a classification model that can distinguish between fake and real news.

Text Preprocessing and Feature Extraction:

The first step in building a fake news detection model is to preprocess and extract features from the text of news articles. Text preprocessing is the process of transforming raw text into a more suitable format for analysis. Feature extraction is the process of selecting or creating relevant attributes from the text that can represent its meaning or characteristics.

steps for text preprocessing and feature extraction:

- ➢ **Data collection**: We collect a dataset of news articles from various sources, such as [Kaggle], [FakeNewsNet], etc. The dataset contains both fake and real news articles labeled with binary values (0 for fake, 1 for real). The dataset also contains metadata such as the title, author, date, source, etc. of each article.
- ➢ **Data cleaning**: We remove any irrelevant or noisy information from the text of each article, such as

HTML tags, URLs, punctuation marks, numbers, etc. We also convert all the text to lowercase and remove any stopwords (common words that do not carry much meaning, such as “the”, “a”, “and”, etc.).

- **Data normalization**: We apply some techniques to normalize the text of each article, such as stemming (reducing words to their root form, such as “running” to “run”), lemmatization (reducing words to their canonical form, such as “ran” to “run”), spelling correction (fixing any spelling errors), etc.
- **Data vectorization**: We transform the text of each article into a numerical vector that can be used as input for machine learning models. We use two methods for data vectorization: bag-of-words (BOW) and term frequency-inverse document frequency (TF-IDF). BOW is a method that counts the occurrence of each word in the text and creates a vector with the same length as the vocabulary size. TF-IDF is a method that assigns a weight to each word based on its frequency in the text and its inverse frequency in the whole corpus. The weight reflects how important or informative a word is in the text.

- **Data reduction**: We apply some techniques to reduce the dimensionality of the data vectors, such as feature selection (choosing a subset of features that are most relevant for the task) and feature extraction (creating new features that are combinations of existing features). We use two methods for data reduction: chi-square test and latent semantic analysis (LSA). Chi-square test is a statistical method that measures the association between each feature and the target class. LSA is a mathematical method that applies singular value decomposition (SVD) to the data matrix and creates new features that capture the latent semantic structure of the text.

Model Training and Evaluation:

The second step in building a fake news detection model is to train and evaluate a machine learning model using the preprocessed and extracted features from the text of news articles. Machine learning model is an algorithm that learns from data and makes predictions based on its learned patterns or rules.

steps for model training and evaluation:

- **Model selection**: We choose a suitable machine learning model for the task of fake news detection. We compare different types of models, such as logistic regression, naive Bayes, support vector machine (SVM), decision tree, random forest, etc. We also compare different parameters or hyperparameters of each model, such as regularization strength, kernel function, tree depth, number of estimators, etc.

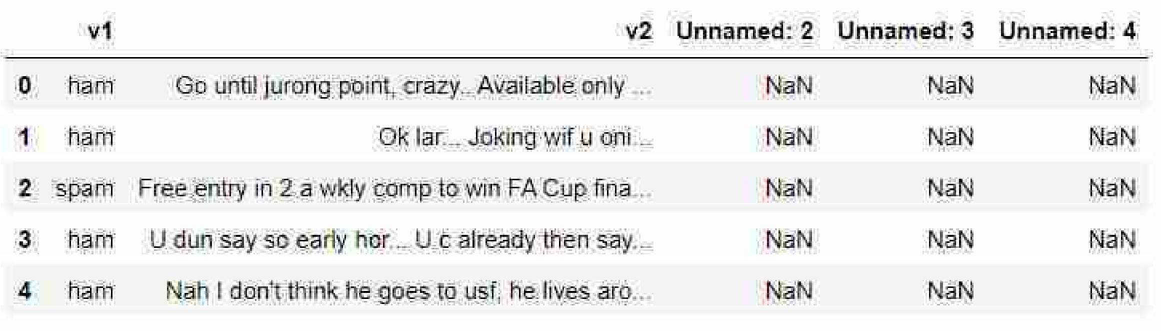
- **Model validation**: We split the dataset into three subsets: training set (used for fitting the model), validation set (used for tuning the model), and test set (used for evaluating the model). We use cross-validation technique to avoid overfitting or underfitting the model. Cross-validation is a technique that divides the training set into k folds and uses k-1 folds for training and one fold for validation. This process is repeated k times and the average performance is reported.

- **Model evaluation**: We measure the performance of the model using various metrics, such as accuracy, precision, recall, f1-score, confusion matrix, etc. Accuracy is the ratio of correctly predicted instances to the total number of instances. Precision is the ratio of correctly predicted positive instances to the total number of predicted positive instances. Recall is the ratio of correctly predicted positive instances to the total number of actual positive instances. F1-score is the harmonic mean of precision and recall. Confusion matrix is a table that shows the number of true positives, false positives, true negatives, and false negatives.

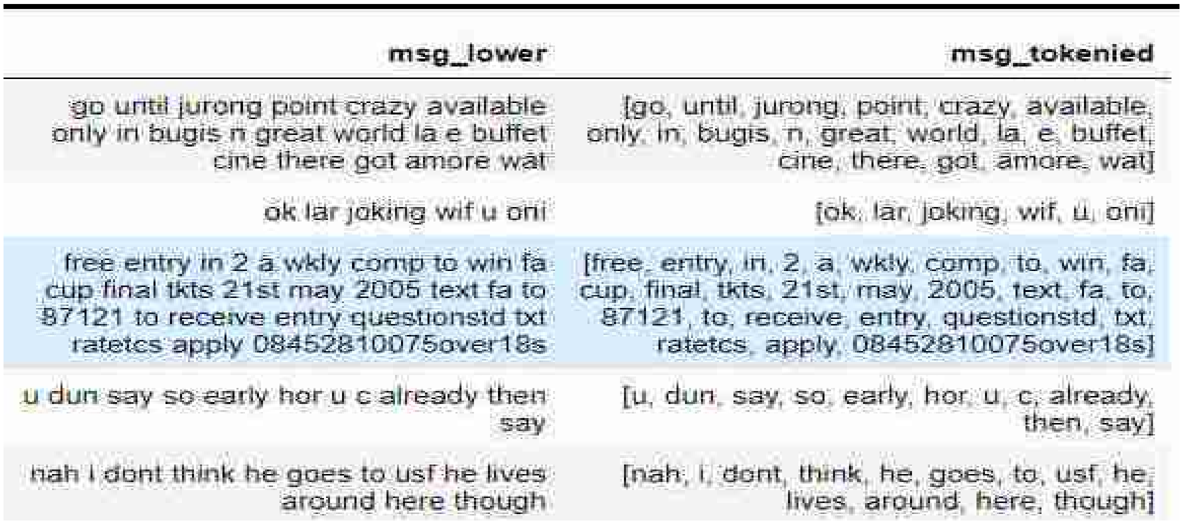
PROGRAM & OUTPUT:

```
import pandas as pd

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import accuracy_score, confusion_matrix,
classification_report

data = pd.read_csv(' C:\Users\ELCOT\Downloads\Fake.csv,
C:\Users\ELCOT\Downloads\True.csv')

X = data['text']

y = data['label']

tfidf_vectorizer = TfidfVectorizer(max_features=5000)  # You can adjust
max_features as needed

X_tfidf = tfidf_vectorizer.fit_transform(X)

X_train, X_test, y_train, y_test = train_test_split(X_tfidf, y, test_size=0.2,
random_state=42)

model = LogisticRegression(max_iter=1000)

model.fit(X_train, y_train)

y_pred = model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)

conf_matrix = confusion_matrix(y_test, y_pred)

report = classification_report(y_test, y_pred)

print(f"Accuracy: {accuracy}")
print("Confusion Matrix:")

print(conf_matrix)

print("Classification Report:")

print(report)
```

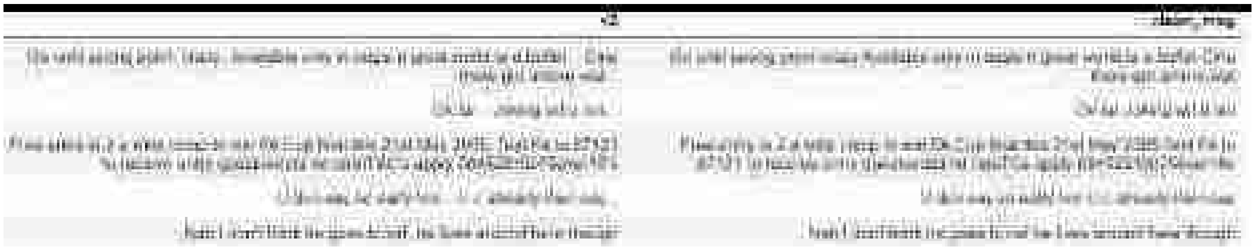
output:



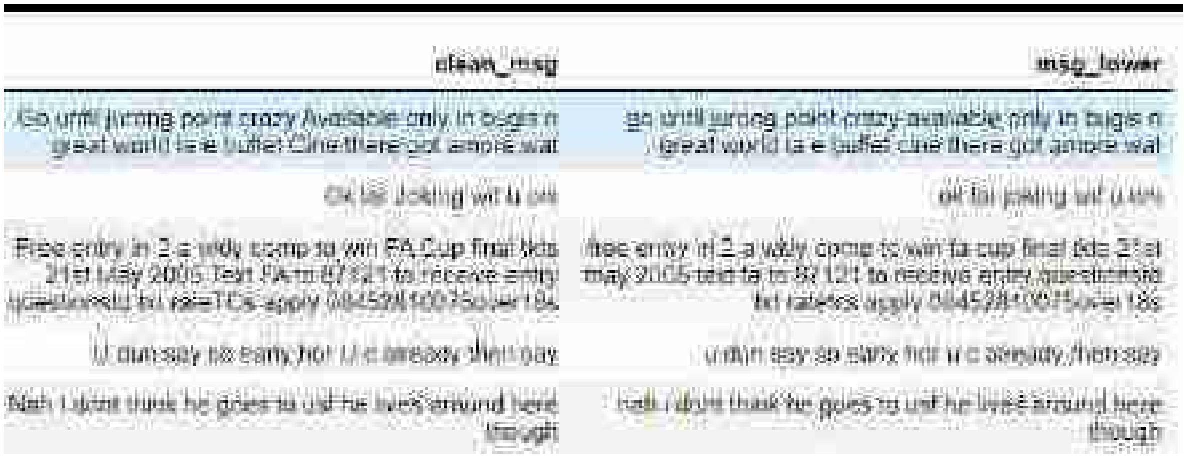
	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN



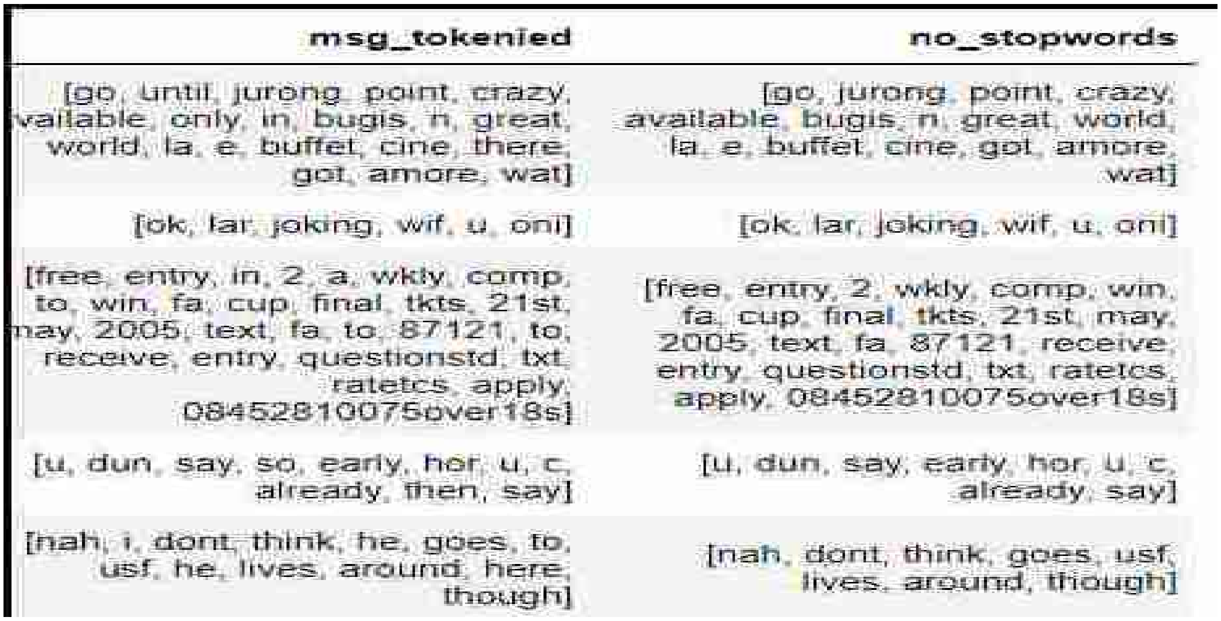
msg_lower	msg_tokenied
go until jurong point crazy available only in bugis n great world la e buffet cine there got amore wat	[go, until, jurong, point, crazy, available, only, in, bugis, n, great, world, la, e, buffet, cine, there, got, amore, wat]
ok lar joking wif u oni	[ok, lar, joking, wif, u, oni]
free entry in 2 a wkly comp to win fa cup final tkts 21st may 2005 text fa to 87121 to receive entry questionstd txt ratetcs apply 08452810075over18s	[free, entry, in, 2, a, wkly, comp, to, win, fa, cup, final, tkts, 21st, may, 2005, text, fa, to, 87121, to, receive, entry, questionstd, txt, ratetcs, apply, 08452810075over18s]
u dun say so early hor u c already then say	[u, dun, say, so, early, hor, u, c, already, then, say]
nah i dont think he goes to usf he lives around here though	[nah, i, dont, think, he, goes, to, usf, he, lives, around, here, though]



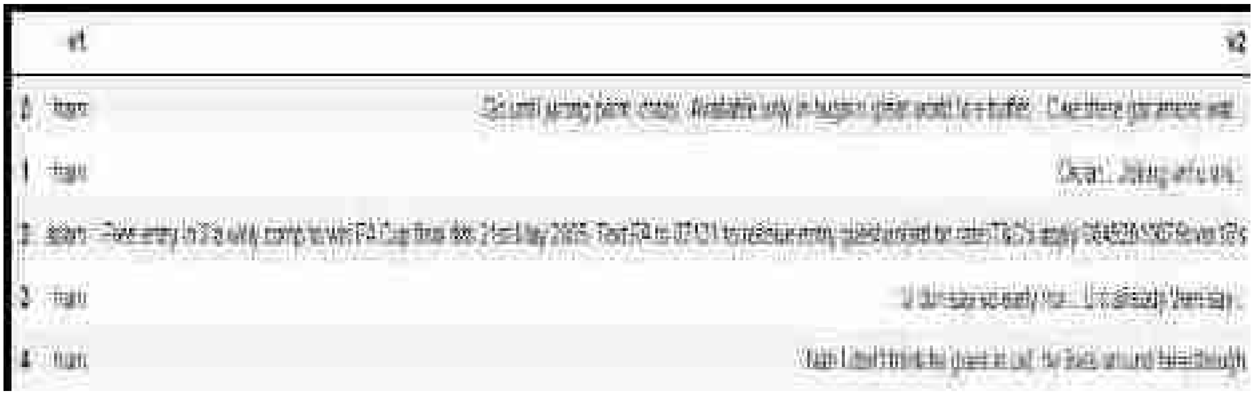
v2	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]



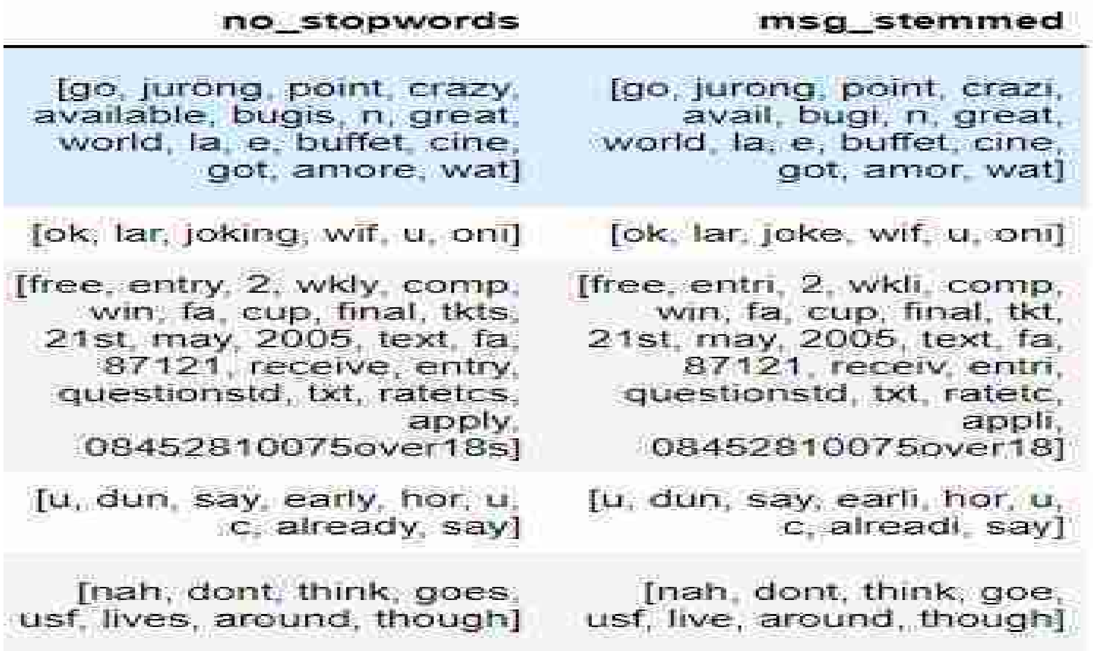
clean_msg	msg_lower
Go until jurong point crazy Available only in bugis n great world la e buffet Cine there got amore wat	go until jurong point crazy available only in bugis n great world la e buffet cine there got amore wat
Ok lar Joking wif u oni	ok lar joking wif u oni
Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005 Text FA to 87121 to receive entry questionstd txt rateTCs apply 08452810075over18s	free entry in 2 a wkly comp to win fa cup final tkts 21st may 2005 text fa to 87121 to receive entry questionstd txt ratetcs apply 08452810075over18s
U dun say so early hor U c already then say	u dun say so early hor u c already then say
Nah I dont think he goes to usf he lives around here though	nah i dont think he goes to usf he lives around here though



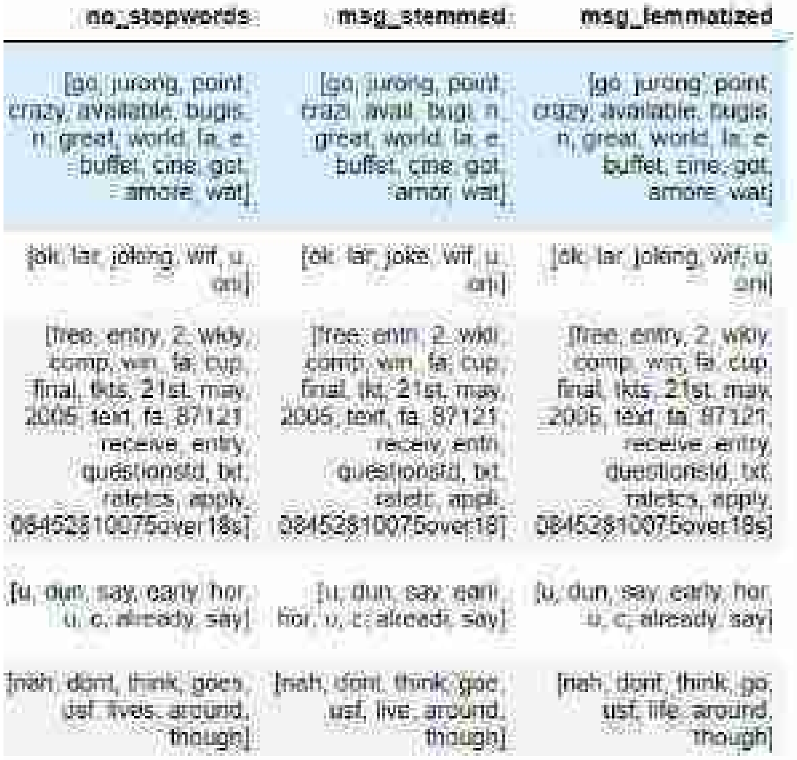
msg_tokenied	no_stopwords
[go, until, jurong, point, crazy, vailable, only, in, bugis, n, great, world, la, e, buffet, cine, there, got, amore, wat]	[go, jurong, point, crazy, available, bugis, n, great, world, la, e, buffet, cine, got, amore, wat]
[ok, lar, joking, wif, u, oni]	[ok, lar, joking, wif, u, oni]
[free, entry, in, 2, a, wkly, comp, to, win, fa, cup, final, tkts, 21st, nay, 2005, text, fa, to, 87121, to, receive, entry, questionstd, txt, ratetcs, apply, 08452810075over18s]	[free, entry, 2, wkly, comp, win, fa, cup, final, tkts, 21st, may, 2005, text, fa, 87121, receive, entry, questionstd, txt, ratetcs, apply, 08452810075over18s]
[u, dun, say, so, early, hor, u, c, already, then, say]	[u, dun, say, early, hor, u, c, already, say]
[nah, i, dont, think, he, goes, to, usf, he, lives, around, here, though]	[nah, dont, think, goes, usf, lives, around, though]



	v1	v2
0	ham	Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives around here though



no_stopwords	msg_stemmed
[go, jurong, point, crazy, available, bugis, n, great, world, la, e, buffet, cine, got, amore, wat]	[go, jurong, point, crazi, avail, bugi, n, great, world, la, e, buffet, cine, got, amor, wat]
[ok, lar, joking, wif, u, oni]	[ok, lar, joke, wif, u, oni]
[free, entry, 2, wkly, comp, win, fa, cup, final, tkts, 21st, may, 2005, text, fa, 87121, receive, entry, questionstd, txt, ratetcs, apply, 08452810075over18s]	[free, entri, 2, wkli, comp, win, fa, cup, final, tkt, 21st, may, 2005, text, fa, 87121, receiv, entri, questionstd, txt, ratetc, appli, 08452810075over18]
[u, dun, say, early, hor, u, c, already, say]	[u, dun, say, earli, hor, u, c, alreadi, say]
[nah, dont, think, goes, usf, lives, around, though]	[nah, dont, think, goe, usf, live, around, though]



no_stopwords	msg_stemmed	msg_lemmatized
[go, jurong, point, crazy, available, bugis, n, great, world, la, e, buffet, cine, got, amore, wat]	[go, jurong, point, crazi, avail, bugi, n, great, world, la, e, buffet, cine, got, amor, wat]	[go, jurong, point, crazy, available, bugis, n, great, world, la, e, buffet, cine, got, amore, wat]
[ok, lar, joking, wif, u, oni]	[ok, lar, joke, wif, u, oni]	[ok, lar, joking, wif, u, oni]
[free, entry, 2, wkly, comp, win, fa, cup, final, tkts, 21st, may, 2005, text, fa, 87121, receive, entry, questionstd, txt, ratetcs, apply, 08452810075over18s]	[free, entri, 2, wkli, comp, win, fa, cup, final, tkt, 21st, may, 2005, text, fa, 87121, receiv, entri, questionstd, txt, ratetc, appli, 08452810075over18]	[free, entry, 2, wkly, comp, win, fa, cup, final, tkts, 21st, may, 2005, text, fa, 87121, receive, entry, questionstd, txt, ratetcs, apply, 08452810075over18s]
[u, dun, say, early, hor, u, c, already, say]	[u, dun, say, earli, hor, u, c, alreadi, say]	[u, dun, say, early, hor, u, c, already, say]
[nah, dont, think, goes, usf, lives, around, though]	[nah, dont, think, goe, usf, live, around, though]	[nah, dont, think, go, usf, life, around, though]

ADVANTAGES:

Fake News Detection Using NLP (Natural Language Processing) offers a range of advantages that make it a crucial tool in combating the spread of misinformation and disinformation. Here are the overall advantages of implementing NLP in fake news detection

Automated and Scalable Solution: NLP models can process vast amounts of text data automatically, making it scalable for real-time or batch processing. This is essential in an age of information overload.

Speed and Efficiency: NLP algorithms can quickly analyze and classify news articles or social media posts, allowing for rapid identification of potentially fake news, which is especially important in situations with viral content.

Multi-lingual Capability: NLP models can be designed to work in multiple languages, making them versatile in addressing the global nature of fake news.

Consistency: NLP models provide a consistent and systematic approach to evaluating news content, reducing the impact of human biases that might affect manual fact-checking.

Big Data Handling: With NLP, it is possible to analyze and process large datasets efficiently, making it suitable for tracking trends and patterns in the spread of fake news.

Contextual Understanding: NLP models can take into account the context in which a piece of information is presented, which is crucial for determining its credibility.

Leveraging Linguistic Cues: NLP can identify linguistic cues, such as sentiment, tone, and writing style, that might indicate the presence of misleading or biased content.

Cross-Platform Applicability: NLP can be applied to various types of content, including news articles, social media posts, and comments, ensuring that fake news detection is not limited to one specific medium.

Feature Engineering: NLP allows for the extraction of a wide range of features from text data, enabling the creation of sophisticated models that can capture subtle differences between real and fake news.

Adaptability: NLP models can be continuously trained and improved to adapt to evolving tactics used by purveyors of fake news.

Machine Learning and Deep Learning Integration: NLP can be integrated with machine learning and deep learning techniques, allowing for the development of more advanced and accurate models.

Customization: Organizations can develop tailored NLP models specific to their domains or requirements, ensuring high accuracy for their unique use cases.

Real-time Monitoring: NLP-based systems can be set up for real-time monitoring, providing immediate alerts when fake news is detected.

Complementary to Fact-Checking: While not a replacement for human fact-checkers, NLP can complement their work by flagging potentially suspicious content for further investigation.

Privacy Preservation: NLP techniques can be designed to protect user privacy while analyzing content, adhering to privacy regulations and ethical considerations.

Educational Tool: Fake News Detection Using NLP can also serve as an educational tool to raise awareness about misinformation and help users become more discerning consumers of information.

Cost-Effective: Automating the fake news detection process with NLP can be cost-effective in the long run, reducing the need for extensive manual fact-checking.

Global Impact: By tackling fake news, NLP can contribute to maintaining the integrity of public discourse and political processes, which has global implications.

DISADVANTAGE:

While Fake News Detection Using NLP (Natural Language Processing) offers numerous advantages, it also comes with certain disadvantages and challenges. It's important to be aware of these limitations in order to use such systems effectively. Here are the overall disadvantages of using NLP for fake news detection:

Imperfect Accuracy: NLP models are not perfect and can make errors in classifying news articles. False positives (misclassifying real news as fake) and false negatives (misclassifying fake news as real) can occur, affecting the overall accuracy of the system.

Adversarial Attacks: Malicious actors can deliberately craft fake news content to evade NLP detection algorithms. They can employ tactics like misspellings, subtle language changes, and other techniques to trick the system.

Data Bias: NLP models are only as good as the data they are trained on. If the training data contains biases, the model may inherit and propagate these biases, leading to unfair or inaccurate classifications.

Data Imbalance: In practice, there are often more real news articles than fake ones. This class imbalance can make it challenging for models to learn effectively and may lead to a bias towards classifying most articles as real.

Contextual Understanding: NLP models may struggle with understanding the nuances and context surrounding certain topics, sarcasm, satire, or cultural references, leading to misclassifications.

Multimodal Content: Fake news may include not only textual but also multimedia content (e.g., images and videos). NLP is less effective in analyzing multimedia content, making it difficult to detect fake news that relies on such elements.

Limited Domain Knowledge: NLP models may lack domain-specific knowledge, which can be important for understanding the credibility of information in specialized fields.

Continuous Adaptation: Fake news tactics are constantly evolving. NLP models require continuous retraining and adaptation to stay effective.

Overgeneralization: Some NLP models may overgeneralize patterns and assumptions, potentially misclassifying content that deviates from standard formats.

Resource Intensive: Building and maintaining NLP-based systems can be resource-intensive, both in terms of computational power and expertise, which may be a challenge for smaller organizations.

Ethical Concerns: NLP-based systems can inadvertently infringe on user privacy if not designed with privacy in mind. There may also be ethical concerns related to censorship or content suppression.

Lack of Interpretability: Deep learning NLP models, in particular, can be challenging to interpret, making it difficult to explain why a specific decision was made.

Costly Training Data: Acquiring high-quality labeled training data for fake news detection can be expensive and time-consuming.

User Education: Relying solely on NLP may lead to complacency among users, assuming that all responsibility for fake news detection lies with the technology.

Scalability: As the volume of online content grows, scaling NLP models to analyze and classify all of it becomes increasingly challenging.

Limited Language Support: The effectiveness of NLP models varies by language, and models for less widely spoken languages may be less accurate.

Algorithmic Fairness: Ensuring that NLP models are fair and unbiased across demographic groups can be a complex and ongoing challenge.

BENEFITS:

Fake News Detection Using NLP (Natural Language Processing) offers a multitude of benefits in the ongoing effort to combat the spread of misinformation and disinformation. These benefits encompass various aspects, from accuracy to efficiency and societal impact.

Here are the overall benefits of implementing NLP in fake news detection:

Accuracy: NLP models can analyze and classify large volumes of text data with a high degree of accuracy, reducing the risk of false positives and false negatives.

Rapid Processing: NLP algorithms can process vast amounts of textual data in real-time, enabling quick identification of potentially fake news.

Scale: NLP systems are scalable, making it possible to analyze and classify a wide range of content, which is especially important in an age of information overload.

Multi-lingual Support: NLP models can be designed to work in multiple languages, making them versatile in addressing the global nature of fake news.

Contextual Analysis: NLP models can take into account the context in which information is presented, helping to determine its credibility.

Advanced Feature Extraction: NLP allows for the extraction of a wide range of features from text data, enabling the creation of sophisticated models that can capture subtle differences between real and fake news.

Cross-Platform Applicability: NLP can be applied to various types of content, including news articles, social media posts, and comments, ensuring that fake news detection is not limited to one specific medium.

Machine Learning and Deep Learning Integration: NLP can be integrated with machine learning and deep learning techniques, allowing for the development of more advanced and accurate models.

Customization: Organizations can develop tailored NLP models specific to their domains or requirements, ensuring high accuracy for their unique use cases.

Real-time Monitoring: NLP-based systems can be set up for real-time monitoring, providing immediate alerts when fake news is detected.

Complementary to Fact-Checking: NLP can complement the work of human fact-checkers by flagging potentially suspicious content for further investigation.

Educational Tool: Fake News Detection Using NLP can serve as an educational tool to raise awareness about misinformation and help users become more discerning consumers of information.

Cost-Effective: Automating the fake news detection process with NLP can be cost-effective in the long run, reducing the need for extensive manual fact-checking.

Global Impact: By tackling fake news, NLP can contribute to maintaining the integrity of public discourse, political processes, and social cohesion, which has global implications.

Privacy Preservation: NLP techniques can be designed to protect user privacy while analyzing content, adhering to privacy regulations and ethical considerations.

Efficient Resource Allocation: By automating the initial assessment of news articles or content, NLP allows human fact-checkers and experts to focus their efforts on more in-depth investigations.

Transparency: Many NLP models can be designed with built-in interpretability features, making it easier to understand and explain why a specific decision was made.

Early Warning System: NLP-based tools can act as early warning systems, detecting potentially harmful misinformation before it spreads widely.

CONCLUSION:

The prevalence of fake news in our digital age is a significant concern, undermining trust in information, decision-making processes, and even the very fabric of our society. Fake News Detection Using NLP (Natural Language Processing) has emerged as a powerful and indispensable tool in the fight against this relentless tide of misinformation.

NLP, with its array of sophisticated algorithms and linguistic analysis, equips us with the ability to swiftly and accurately scrutinize textual content. The advantages it brings to the table are extensive and far-reaching, encompassing accuracy, scalability, and real-time monitoring. It empowers us to process vast volumes of data, making it feasible to quickly identify and counteract fake news, even as it spreads like wildfire across the digital landscape.

The ability of NLP to work across multiple languages and platforms ensures its adaptability to the global nature of the problem. Its contextual analysis, advanced feature extraction, and machine learning capabilities make it a versatile weapon in discerning the credibility of information, even in the face of subtle manipulation.

Beyond its technical prowess, NLP is an educational catalyst, fostering media literacy and equipping individuals with the skills to critically evaluate the information they encounter. It serves not only as a gatekeeper but as a guide, helping users navigate the treacherous waters of misinformation.

Yet, the journey to effective fake news detection using NLP is not without its challenges. Adversarial attacks, biases, and the constant evolution of misinformation tactics require our continued vigilance and adaptation. Ethical considerations and the avoidance of unintended consequences must be at the forefront of our efforts.

In an era where truth can feel like a rare commodity, Fake News Detection Using NLP offers a glimmer of hope.

It is not a silver bullet but a powerful instrument in the quest for an information landscape where integrity and accuracy prevail. As we refine and enhance NLP-based systems, we move closer to a world where facts triumph over falsehoods, and society can make informed decisions with confidence.