PHASE 2 : INNOVATION PHASE

TEAM MEMBERS

513121106061

513121106094

513121106097

513121106312

513121106310

COURSE	Artificial intelligence
PROJECT	Fake News detection using NLP
DATE	11.10.2023

Project : Fake News Detection Using NLP

INTRODUCTION:

- In today's digital era, over 50% of readers are e-Readers. Fast response time, low cost, and a large Data storage capacity makes the internet a popular Source of news and information. Today, it is one of The most important sources of knowledge in people's Lives. As the term implies, fake news is inaccurate And specious information masquerading as news. Besides hurting people's feelings, it may cause Damage to their image, mislead public opinion, or Cause major conflicts. In some cases, authors and Websites lurk at people in order to monetize their Content or gain media coverage by using their Influence and clickbait. So, it is reasonable to Consider them as one of the most significant Menaces to community and confraternity. As the Internet and social communication sites have Become more widely available, the rate of Generating this fake news has increased Dramatically. This news is produced in bulk which Makes it difficult to detect in real-time analysis.

- Since the world population has been expanding on a Massive scale from the past decade, it is robustly Important for people to fathom actual authenticated And hoax news.

CONTENT FOR PHASE 2

Consider collaboration with fact-checking organizations can provide valuable input to improve the accuracy of detection.

STATEMENT

In this phase, we can explore innovative techniques such as ensemble methods and deep Learning architectures to improve the detection system's accuracy and robustness.

DESCRIPTION OF THE DATASET TOPIC

- FND-2023 is a curated dataset collected from various online sources, including social media platforms, news websites, and fact-checking organizations.
- The data collection process involved web scraping, API access, and manual curation to ensure a diverse and representative sample of news articles.
- FND-2023 contains a total of 50,000 news articles, evenly split between real and fake articles.
- The dataset is balanced to ensure an equal representation of both classes.
- Articles are diverse in terms of topics, sources, and writing styles to mimic real-world conditions.

MODEL SELECTION AND EVALUATION

The quality of dataset annotations is paramount. High-quality annotations, ideally from reputable fact-checking organizations or experts, are essential for training accurate models. Additionally, consider ethical considerations, ensuring the dataset adheres to guidelines and avoids harmful or offensive content.

MODEL INTERPRETABILITY

- Explain how a machine learning model makes predictions.
- It's essential for building trust in AI systems, ensuring accountability, and diagnosing and improving model performance.

DEPLOYMENT AND PREDICTION

- Choose the best-performing model based on validation results. Consider factors like accuracy, interpretability, and computational efficiency.
- Develop data preprocessing pipelines that mimic those used during model training. Ensure that input data is transformed and scaled consistently.
- Collect user feedback on model predictions and use it to improve model accuracy and user satisfaction through iterative updates.

PROGRAM

Fake News detection using NLP

In [1]:

```
Import warnings
Warnings.filterwarnings('ignore')
```

In [2]:

```
Import numpy as np
Import pandas as pd
Import seaborn as sns
Import matplotlib.pyplot as plt
Import nltk
From sklearn.preprocessing import LabelBinarizer
From nltk.corpus import stopwords
From nltk.stem.porter import PorterStemmer
From wordcloud import WordCloud,STOPWORDS
From nltk.stem import WordNetLemmatizer
From nltk.tokenize import word_tokenize,sent_tokenize
From bs4 import BeautifulSoup
Import re,string,unicodedata
From nltk import pos_tag
From nltk.corpus import wordnet
From sklearn.metrics import
classification_report,confusion_matrix,accuracy_score
From sklearn.model_selection import train_test_split
```

From string import punctuation

In [3]:

Import tensorflow as tf

Import keras

From keras.preprocessing import text, sequence

From keras.utils import pad_sequences

From keras.models import Sequential

From keras.layers import Dense,Embedding,LSTM,Dropout

From keras.callbacks import ReduceLROnPlateau

IMPORTING THE DATASET

In[4]:

True = pd.read_csv("../input/fake-and-real-news-dataset/True.csv")

False = pd.read_csv("../input/fake-and-real-news-dataset/Fake.csv")

DATA VISUALIZATION AND PREPROCESSING

In[5]:

true.head()

Out:

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

In [6]:

False.head()

Out[6]:

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

In[7]

True['category'] = 1

False['category'] = 0

In[8]:

#Merging the 2 datasets

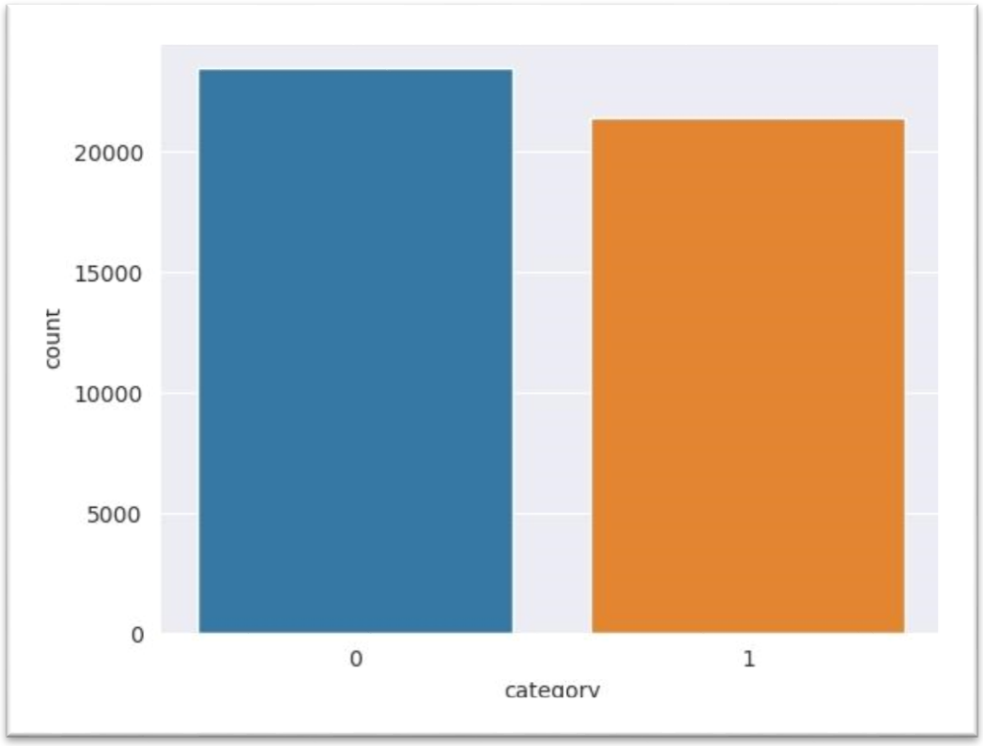
Df = pd.concat([true,false])

In[9]:

Sns.set_style("darkgrid")

Sns.countplot(df , x = 'category')

Out[9]:



In[10]:

Df.head()

Out[10]:

	title	text	subject	date	categ
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[11]:

Checking for nan Values

Df.isna().sum()

Out[11]:

Title 0

Text 0

Subject 0

Date 0

Category 0

Dtype: int64

In[12]:

Df.title.count()

Out[12]:

44898

In[13]:

Df.subject.value_counts()

Out[13]:

politicsNews 11272

worldnews 10145

News 9050

Politics 6841

Left-news 4459

Government News 1570

US_News 783

Middle-east 778

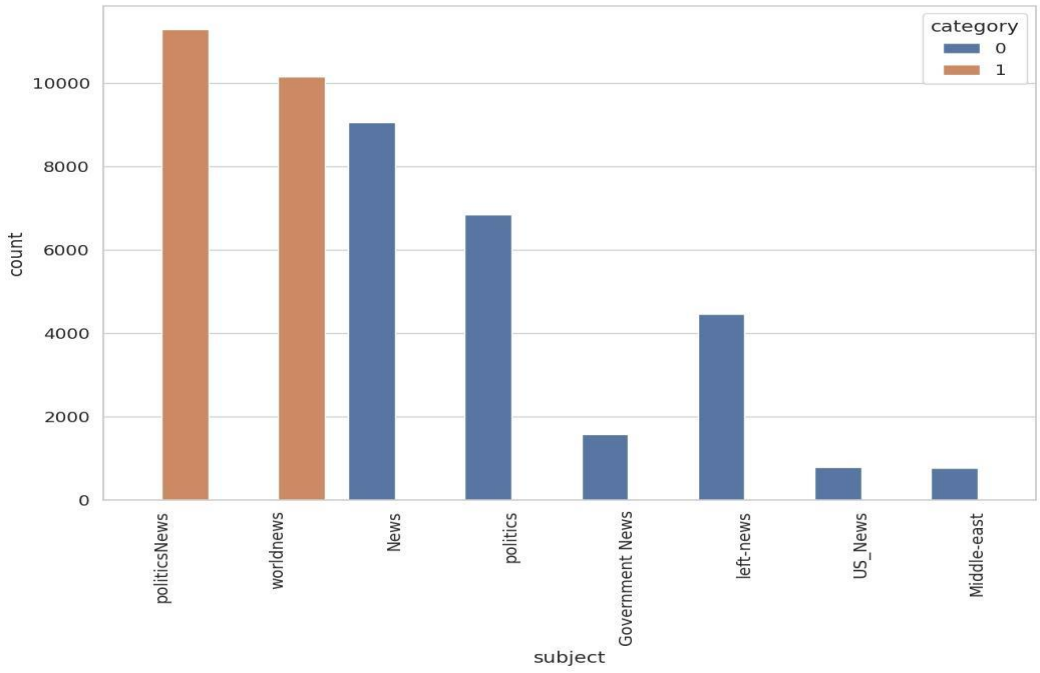
Name: subject, dtype: int64

In[14]:

```
Plt.figure(figsize = (12,8))
Sns.set(style = “whitegrid”,font_scale = 1.2)
Chart = sns.countplot(x = “subject”, hue = “category” , data = df)
Chart.set_xticklabels(chart.get_xticklabels(),rotation=90
```

Out[14]:

```
[Text(0, 0, ‘politicsNews’),
 Text(1, 0, ‘worldnews’),
 Text(2, 0, ‘News’),
 Text(3, 0, ‘politics’),
 Text(4, 0, ‘Government News’),
 Text(5, 0, ‘left-news’),
 Text(6, 0, ‘US_News’),
 Text(7, 0, ‘Middle-east’)]
```



In[15]:

Df['text'] = df['text'] + " " + df['title']

Del df['title']

Del df['subject']

Del df['date']

In[16]:

Stop = set(stopwords.words('english'))

Punctuation = list(string.punctuation)

Stop.update(punctuation)

In[17]:

#Removing the square brackets

Def remove_between_square_brackets(text):

    Return re.sub('(\[[^]]*\])', '', text)

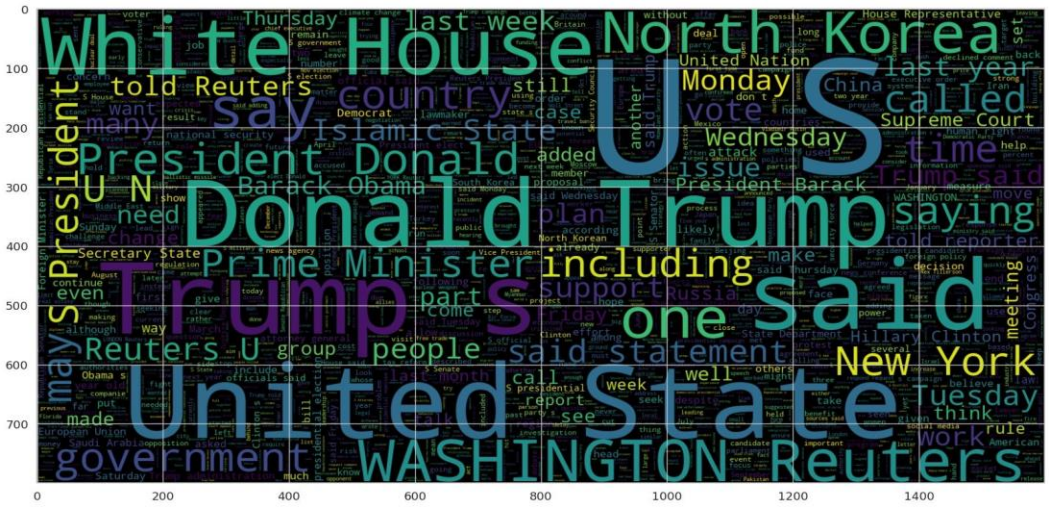
```
# Removing URL's
Def remove_between_square_brackets(text):
    Return re.sub(r'http\S+', '', text)
#Removing the stopwords from text
Def remove_stopwords(text):
    Final_text = []
    For I in text.split():
        If i.strip().lower() not in stop:
            Final_text.append(i.strip())
    Return " ".join(final_text)
#Removing the noisy text
Def denoise_text(text):
    Text = remove_between_square_brackets(text)
    Text = remove_stopwords(text)
    Return text
#Apply function on review column
Df['text']=df['text'].apply(denoise_text)
```

In[18]:

```
Plt.figure(figsize = (20,20)) # Text that is not Fake
Wc = WordCloud(max_words = 2000 , width = 1600 , height = 800 , stopwords =
STOPWORDS).generate(" ".join(df[df.category == 1].text))
Plt.imshow(wc , interpolation = 'bilinear')
```

Out[18]:

<matplotlib.image.AxesImage at 0x7f59e986cb80>



In[19]:

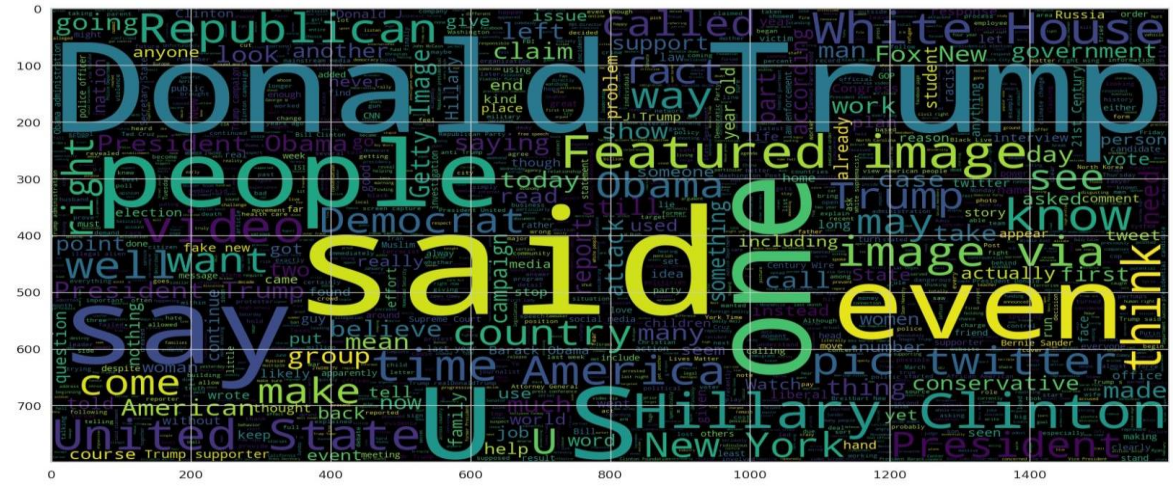
Plt.figure(figsize = (20,20)) # Text that is Fake

Wc = WordCloud(max_words = 2000 , width = 1600 , height = 800 , stopwords = STOPWORDS).generate(“ “.join(df[df.category == 0].text))

Plt.imshow(wc , interpolation = ‘bilinear’)

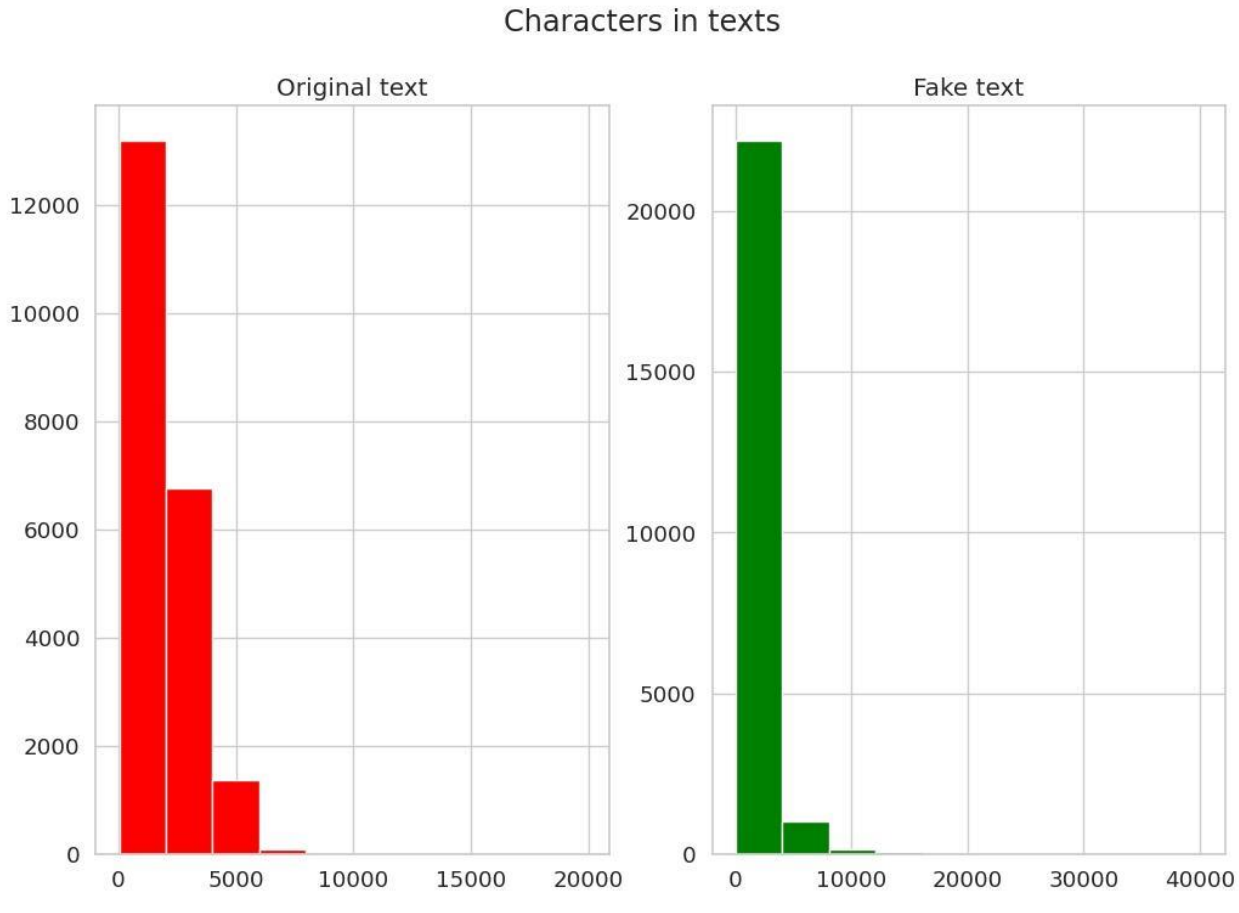
Out[19]:

<matplotlib.image.AxesImage at 0x7f59e49e6800>



In[20]:

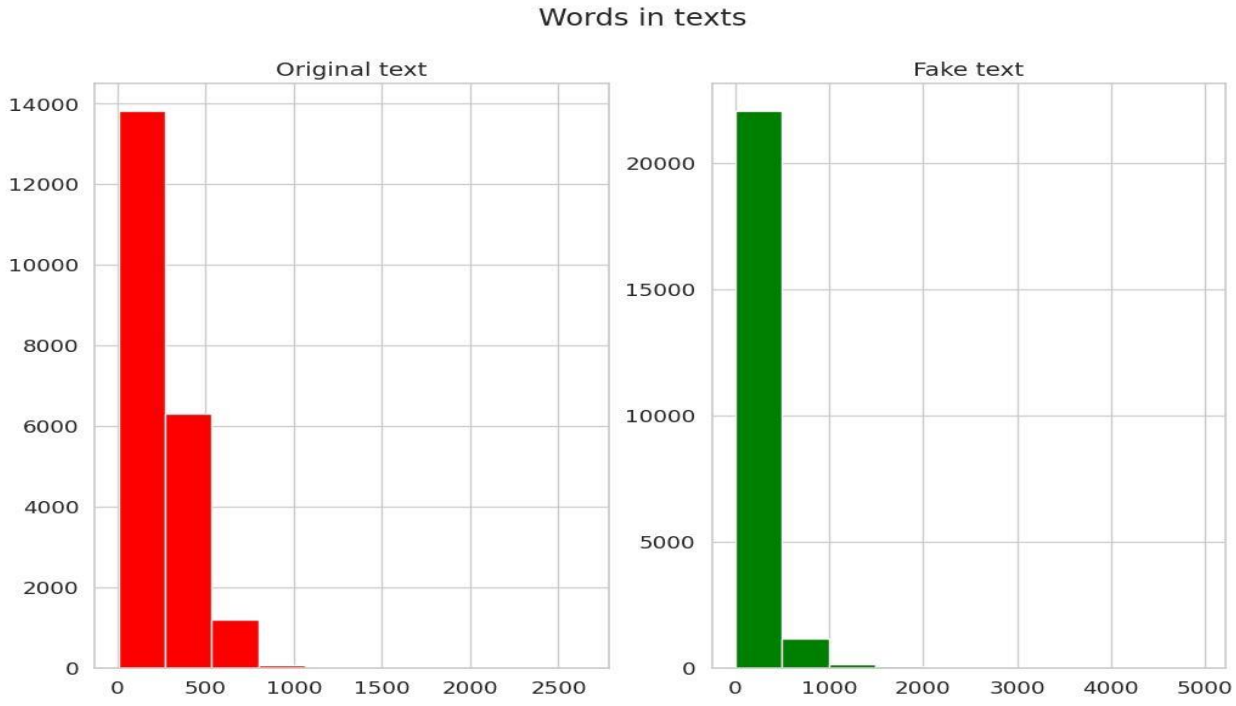
```
Fig,(ax1,ax2)=plt.subplots(1,2,figsize=(12,8))
Text_len=df[df['category']==1]['text'].str.len()
Ax1.hist(text_len,color='red')
Ax1.set_title('Original text')
Text_len=df[df['category']==0]['text'].str.len()
Ax2.hist(text_len,color='green')
Ax2.set_title('Fake text')
Fig.suptitle('Characters in texts')
Plt.show()
```



Number of words in each text

In[21]:

```
Fig,(ax1,ax2)=plt.subplots(1,2,figsize=(12,8))
Text_len=df[df['category']==1]['text'].str.split().map(lambda x: len(x))
Ax1.hist(text_len,color='red')
Ax1.set_title('Original text')
Text_len=df[df['category']==0]['text'].str.split().map(lambda x: len(x))
Ax2.hist(text_len,color='green')
Ax2.set_title('Fake text')
Fig.suptitle('Words in texts')
Plt.show()
```



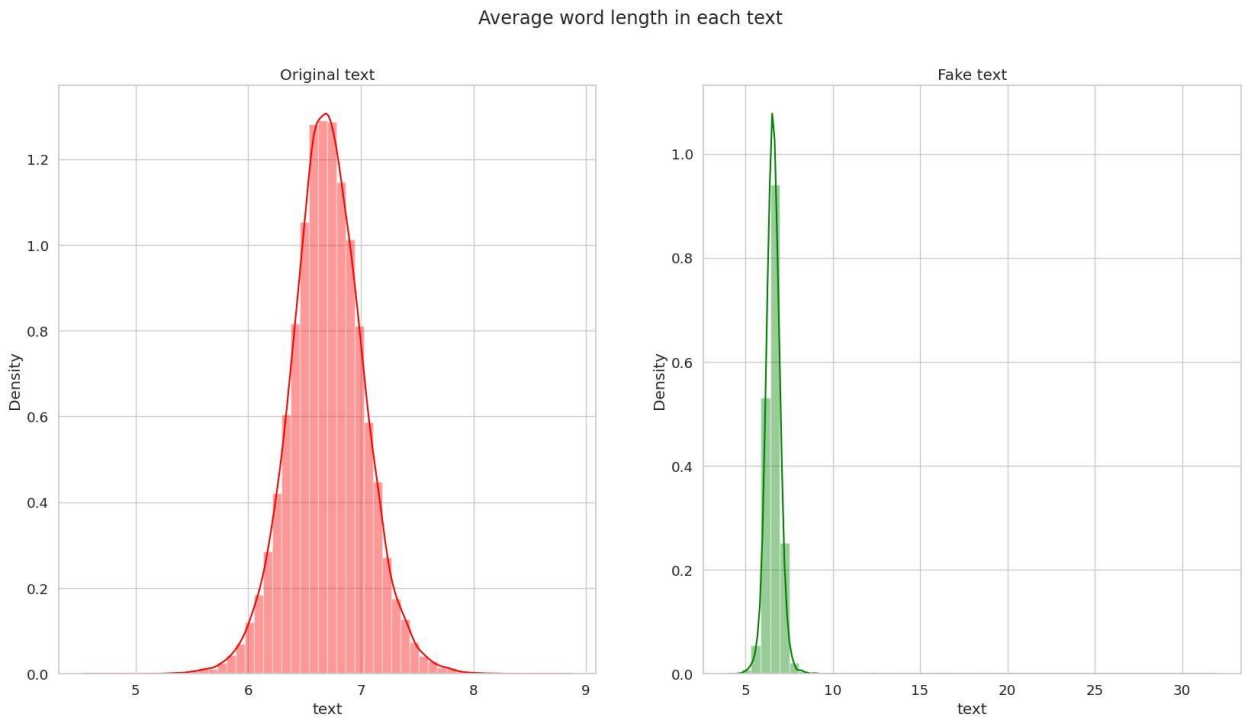
Average word length in a text

In[22]:

```
Fig,(ax1,ax2)=plt.subplots(1,2,figsize=(20,10))
Word=df[df['category']==1]['text'].str.split().apply(lambda x : [len(i) for I in x])
Sns.distplot(word.map(lambda x: np.mean(x)),ax=ax1,color='red')
Ax1.set_title('Original text')
Word=df[df['category']==0]['text'].str.split().apply(lambda x : [len(i) for I in x])
Sns.distplot(word.map(lambda x: np.mean(x)),ax=ax2,color='green')
Ax2.set_title('Fake text')
Fig.suptitle('Average word length in each text')
```

Out[22]:

Text(0.5, 0.98, 'Average word length in each text')



In[23]:

```
Def get_corpus(text):
    Words = []
    For I in text:
        For j in i.split():
            Words.append(j.strip())
    Return words
Corpus = get_corpus(df.text)
Corpus[:5]
```

Out[23]:

['WASHINGTON', '(Reuters)', 'head', 'conservative', 'Republican']

In[24]:

```
From collections import Counter
Counter = Counter(corpus)
Most_common = counter.most_common(10)
Most_common = dict(most_common)
Most_common
```

Out[24]:

```
{'Trump': 111503,
 'said': 93162,
 'would': 54613,
 'U.S.': 50441,
 'President': 33180,
 'people': 33115,
```

‘also’: 30325,

‘one’: 29370,

‘Donald’: 27795,

‘said.’: 26194}

In[25]:

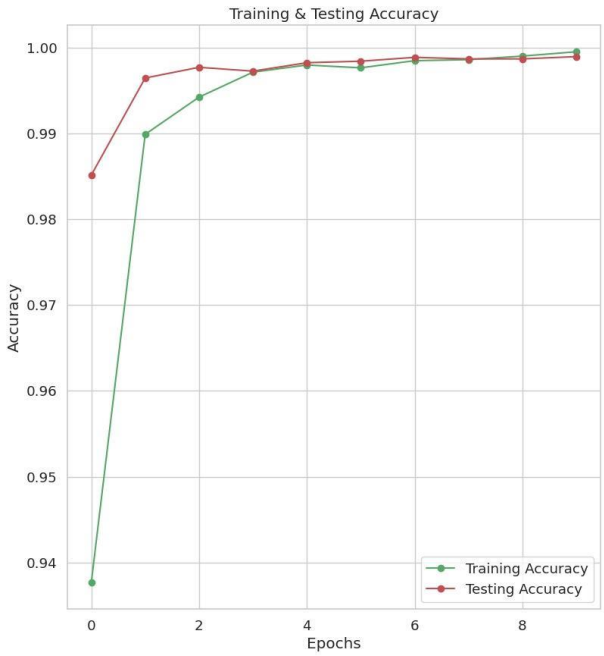
```
Epochs = [I for I in range(10)]
Fig , ax = plt.subplots(1,2)
Train_acc = history.history[‘accuracy’]
Train_loss = history.history[‘loss’]
Val_acc = history.history[‘val_accuracy’]
Val_loss = history.history[‘val_loss’]
Fig.set_size_inches(20,10)

Ax[0].plot(epochs , train_acc , ‘go-‘ , label = ‘Training Accuracy’)
Ax[0].plot(epochs , val_acc , ‘ro-‘ , label = ‘Testing Accuracy’)
Ax[0].set_title(‘Training & Testing Accuracy’)
Ax[0].legend()
Ax[0].set_xlabel(“Epochs”)
Ax[0].set_ylabel(“Accuracy”)

Ax[1].plot(epochs , train_loss , ‘go-‘ , label = ‘Training Loss’)
Ax[1].plot(epochs , val_loss , ‘ro-‘ , label = ‘Testing Loss’)
Ax[1].set_title(‘Training & Testing Loss’)
```

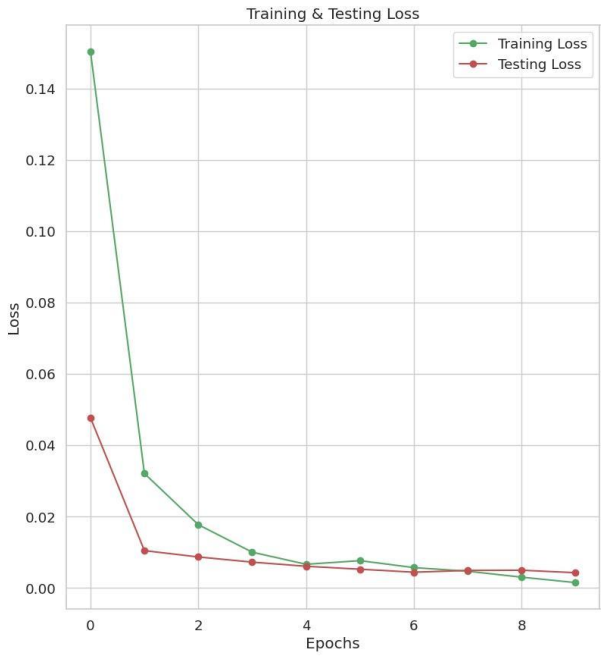
Ax[1].legend()

Ax[1].set_xlabel(“Epochs”)

Ax[1].set_ylabel(“Loss”)

Plt.show()





Conclusion:

In conclusion, our Phase 2 submission outlines a comprehensive and innovative approach to tackle the critical issue of fake news through advanced NLP techniques. Our proposed solution, with its unique methodology and potential impact on information integrity, is well-positioned to make a significant difference.