Description

Develop a machine learning program to identify when an article might be fake news. Run by the UTK Machine Learning Club. The evaluation metric for this competition is accuracy, a very straightforward metric.

$$accuracy = \frac{correct\ predictions}{correct\ predictions + incorrect\ predictions}$$

Accuracy measures false positives and false negeatives equally, and really should only be used in simple cases and when classes are of (generally) equal class size

Dataset Description

train.csv: A full training dataset with the following attributes:

- id: unique id for a news article
- title: the title of a news article
- author: author of the news article
- text: the text of the article; could be incomplete
- label: a label that marks the article as potentially unreliable
  - 1: unreliable
  - 0: reliable

test.csv: A testing training dataset with all the same attributes at train.csv without the label.

Getting Data

In [ ]:
```
import pandas as pd
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [ ]:
```
df_train = pd.read_csv('train.csv')
df_test = pd.read_csv('test.csv')
```

Explore the missing values

In [ ]:
```
def explore_data(df):
    '''Input- df= pandas dataframes to be explored
       Output- print shape, info and first 5 records of the dataframe
    '''
    print("-"*50)
    print('Shape of the dataframe:',df.shape)
    print("Number of records in train data set:",df.shape[0])
    print("Information of the dataset:")
    df.info()
    print("-"*50)
    print("First 5 records of the dataset:")
    return df.head()
    print("-"*50)
```

In [ ]:
```
explore_data(df_train)
```

```
--------------------------------------------------
Shape of the dataframe: (20800, 5)
Number of records in train data set: 20800
Information of the dataset:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20800 entries, 0 to 20799
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   id      20800 non-null  int64
 1   title   20242 non-null  object
 2   author  18843 non-null  object
 3   text    20761 non-null  object
 4   label   20800 non-null  int64
dtypes: int64(2), object(3)
memory usage: 812.6+ KB
--------------------------------------------------
First 5 records of the dataset:
```

Out[ ]:

	id	title	author	text	label
0	0	House Dem Aide: We Didn't Even See Comey's Let...	Darrell Lucus	House Dem Aide: We Didn't Even See Comey's Let...	1
1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	Ever get the feeling your life circles the rou...	0
2	2	Why the Truth Might Get You Fired	Consortiumnews.com	Why the Truth Might Get You Fired October 29, ...	1
3	3	15 Civilians Killed In Single US Airstrike Hav...	Jessica Purkiss	Videos 15 Civilians Killed In Single US Airstr...	1
4	4	Iranian woman jailed for fictional unpublished...	Howard Portnoy	Print \nAn Iranian woman has been sentenced to...	1

In [ ]:
```
explore_data(df_test)
```

```
--------------------------------------------------
Shape of the dataframe: (5200, 4)
Number of records in train data set: 5200
Information of the dataset:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5200 entries, 0 to 5199
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   id      5200 non-null   int64
 1   title   5078 non-null   object
 2   author  4697 non-null   object
 3   text    5193 non-null   object
dtypes: int64(1), object(3)
memory usage: 162.6+ KB
--------------------------------------------------
First 5 records of the dataset:
```

Out[ ]:

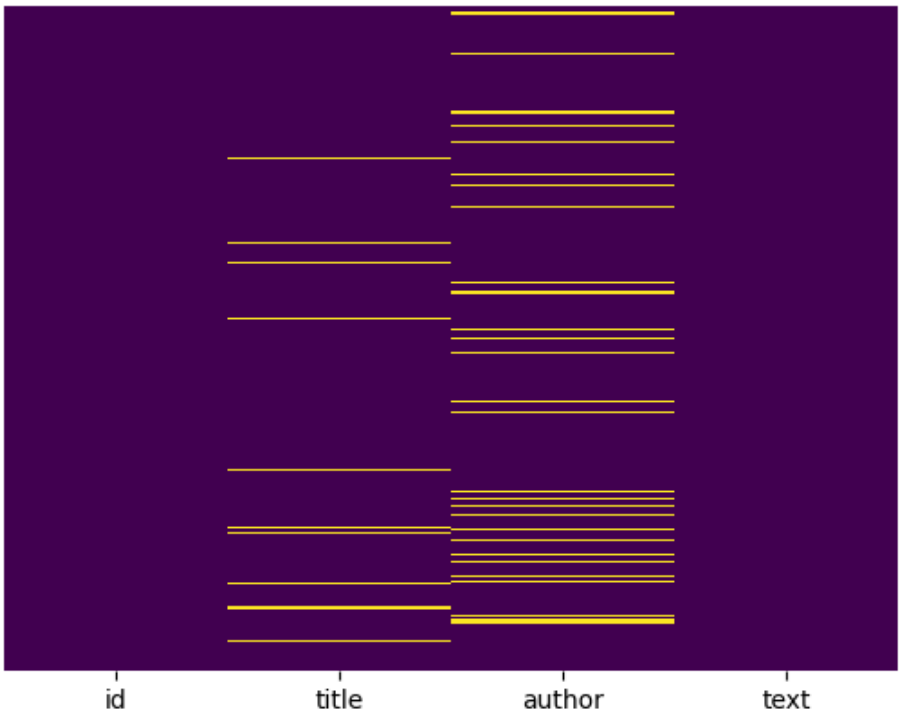
	id	title	author	text
0	20800	Specter of Trump Loosens Tongues, if Not Purse...	David Streitfeld	PALO ALTO, Calif. — After years of scorning...
1	20801	Russian warships ready to strike terrorists ne...	NaN	Russian warships ready to strike terrorists ne...
2	20802	#NoDAPL: Native American Leaders Vow to Stay A...	Common Dreams	Videos #NoDAPL: Native American Leaders Vow to...
3	20803	Tim Tebow Will Attempt Another Comeback, This ...	Daniel Victor	If at first you don't succeed, try a different...
4	20804	Keiser Report: Meme Wars (E995)	Truth Broadcast Network	42 mins ago 1 Views 0 Comments 0 Likes 'For th...

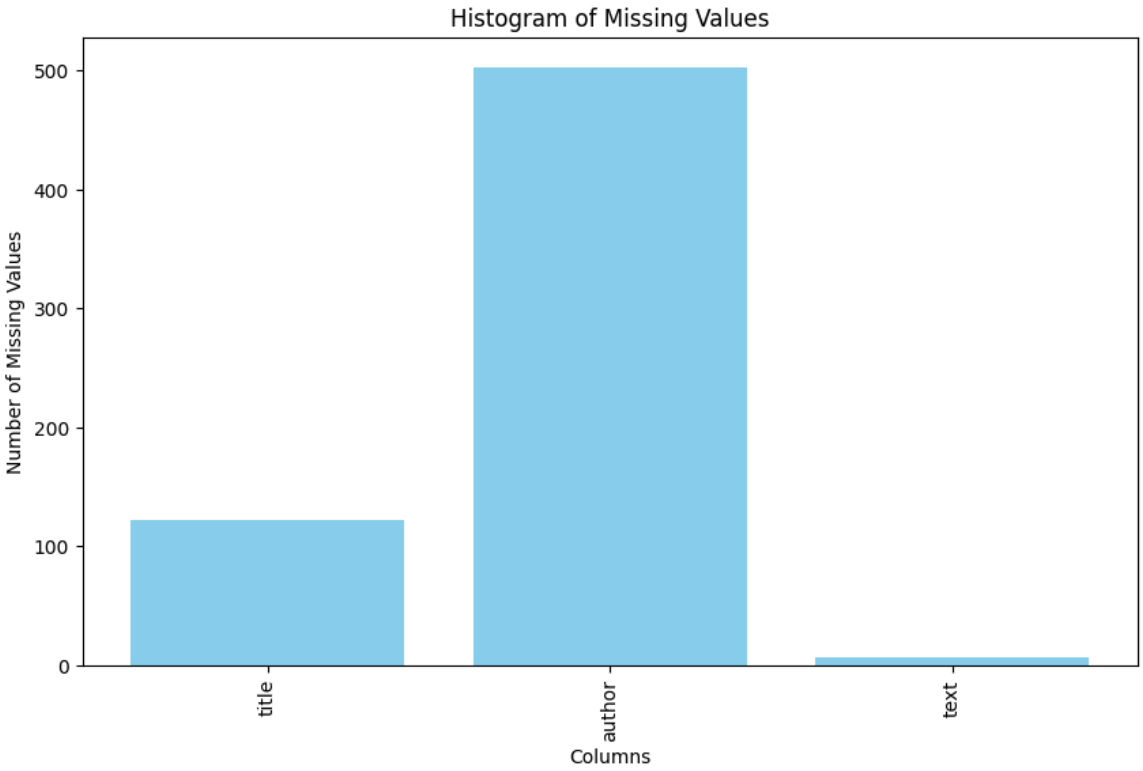
In [ ]:
```
#Let's define a function to explore the missing values for the two datasets

def missing_values(df):
```

```
    print('{}% of title values are missing from Total Number of Records.'.format(round((d
    print('{}% of author values are missing from Total Number of Records.'.format(round((
    print('{}% of text values are missing from Total Number of Records.'.format(round((df

    sns.heatmap(df.isnull(),yticklabels=False,cbar=False,cmap='viridis')

    # Plot histogram of missing values
    missing_counts = df.isnull().sum()
    missing_counts = missing_counts[missing_counts > 0]
    plt.figure(figsize=(10, 6))
    plt.bar(missing_counts.index, missing_counts.values, color='skyblue')
    plt.title('Histogram of Missing Values')
    plt.xlabel('Columns')
    plt.ylabel('Number of Missing Values')
    plt.xticks(rotation=90)
    plt.show()
```

In [ ]:
```
#let's use the missing_values function to see the missing values in the train dataset
missing_values(df_test)
```

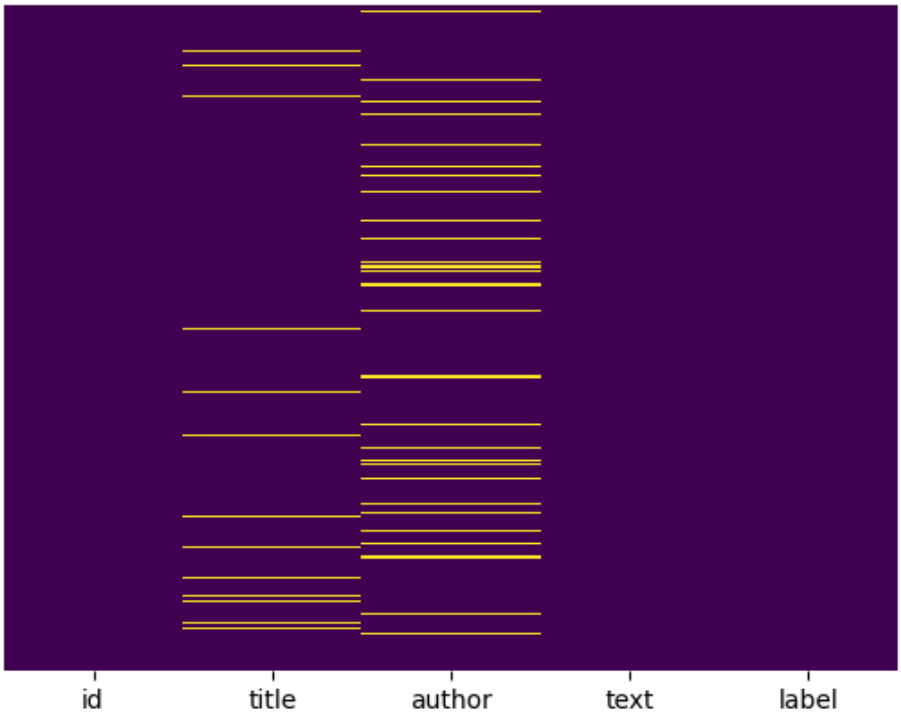
```
2% of title values are missing from Total Number of Records.
10% of author values are missing from Total Number of Records.
0% of text values are missing from Total Number of Records.
```

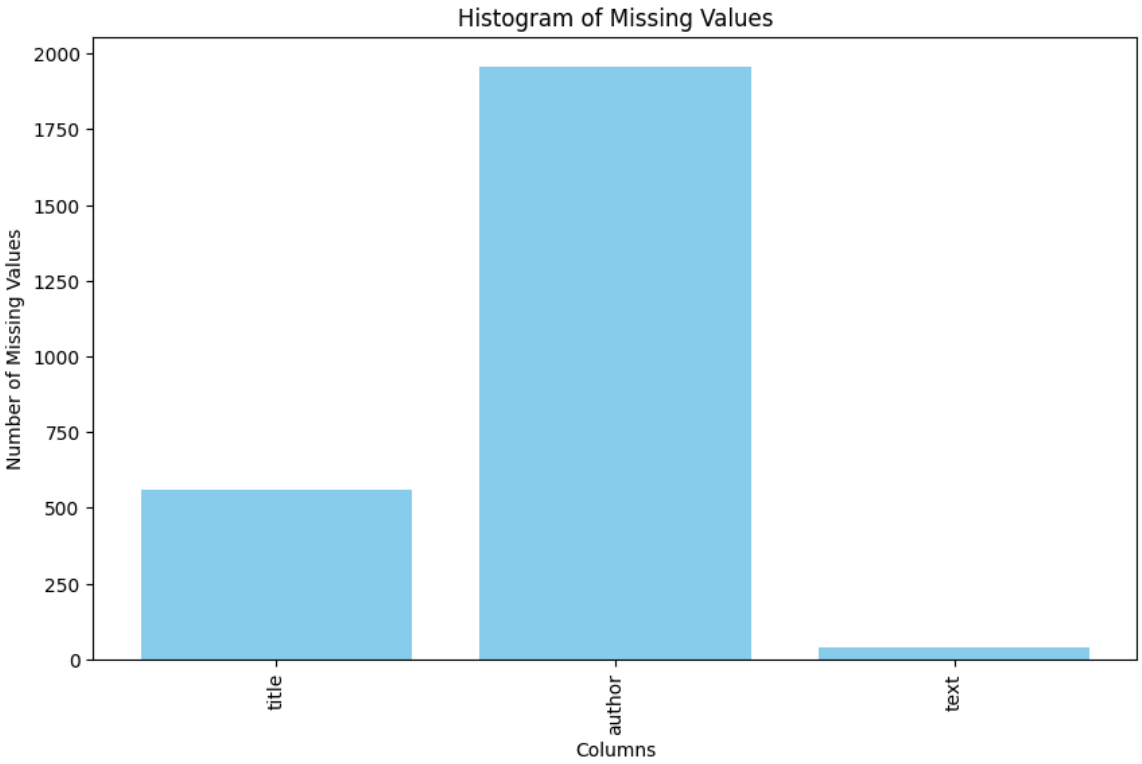




In [ ]:
```
#let's use the missing_values function to see the missing values in the train dataset
missing_values(df_train)
```

```
3% of title values are missing from Total Number of Records.
9% of author values are missing from Total Number of Records.
0% of text values are missing from Total Number of Records.
```





In [ ]:
```
df_test = df_test.dropna()
df_train = df_train.dropna()
```

As our dataset is very extensive we can afford to drop null instances

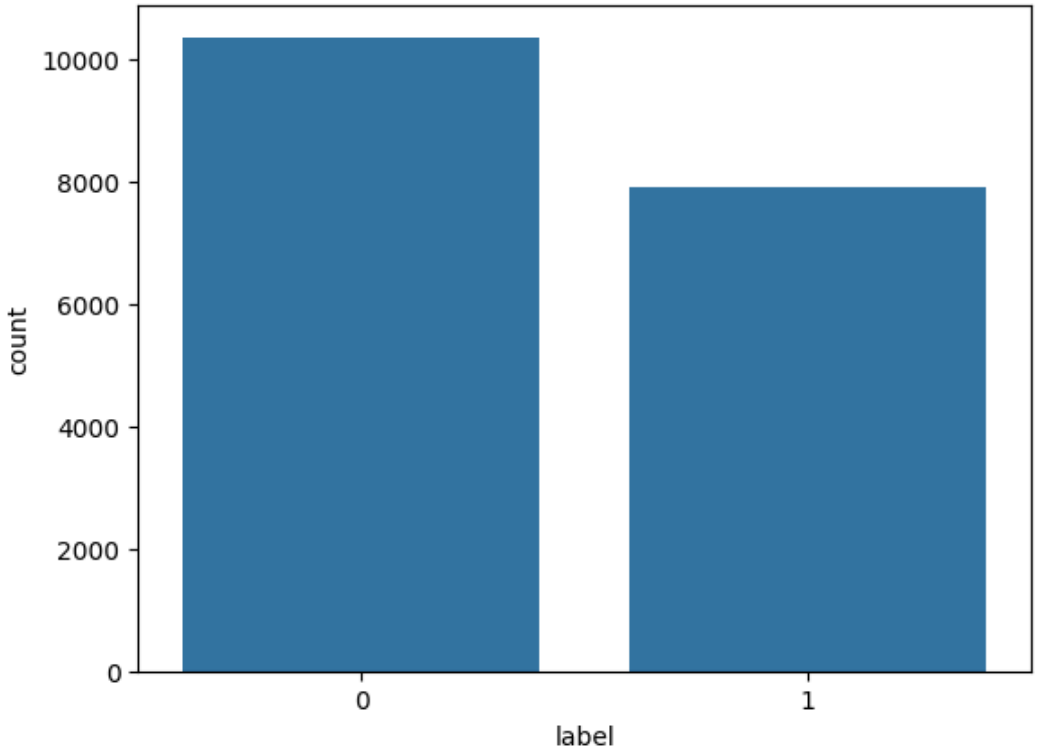
In [ ]:
```
df_train.shape, df_test.shape
```

Out[ ]:
```
((18285, 5), (4575, 4))
```

Class distribution

In [ ]:
```
print('Target of 0 is {} % of total'.format(round(df_train['label'].value_counts()[0]/len
print('Target of 1 is {} % of total'.format(round(df_train['label'].value_counts()[1]/len

x=df_train.label.value_counts()
sns.barplot(x=x.index, y=x)
plt.show()
```

```
Target of 0 is 57 % of total
Target of 1 is 43 % of total
```



Cleaning the Data

Before starting any NLP project, text data needs to be pre-processed to convert it into in a consistent format.Text will be cleaned, tokneized and converted into a matrix.

- Step 1: Lowercase
- Step 2: Punctuation Removal
- Step 3: HTML Code and URL Links
- Step 4: Spell Checks
- Step 5: Tokenization
- Step 6: Removing Stop Words
- Step 7: Normalization
  - Stemming
  - Lemmatization

Step 1: Lowercase

In [ ]:
```
df_train = df_train.applymap(lambda x: x.lower() if isinstance(x, str) else x)
df_train
```

```
C:\Users\sa\AppData\Local\Temp\ipykernel_10936\3467437343.py:1: FutureWarning: DataFrame.
applymap has been deprecated. Use DataFrame.map instead.
  df_train = df_train.applymap(lambda x: x.lower() if isinstance(x, str) else x)
```

Out[ ]:

	id	title	author	text	label
0	0	house dem aide: we didn't even see comey's let...	darrell lucus	house dem aide: we didn't even see comey's let...	1
1	1	flynn: hillary clinton, big woman on campus - ...	daniel j. flynn	ever get the feeling your life circles the rou...	0
2	2	why the truth might get you fired	consortiumnews.com	why the truth might get you fired october 29, ...	1
3	3	15 civilians killed in single us airstrike hav...	jessica purkiss	videos 15 civilians killed in single us airstr...	1
4	4	iranian woman jailed for fictional unpublished...	howard portnoy	print \nan iranian woman has been sentenced to...	1
...	...	...	...	...	...
20795	20795	rapper t.i.: trump a 'poster child for white s...	jerome hudson	rapper t. i. unloaded on black celebrities who...	0
20796	20796	n.f.l. playoffs: schedule, matchups and odds -...	benjamin hoffman	when the green bay packers lost to the washing...	0
20797	20797	macy's is said to receive takeover approach by...	michael j. de la merced and rachel abrams	the macy's of today grew from the union of sev...	0
20798	20798	nato, russia to hold parallel exercises in bal...	alex ansary	nato, russia to hold parallel exercises in bal...	1
20799	20799	what keeps the f-35 alive	david swanson	david swanson is an author, activist, journa...	1

18285 rows × 5 columns

In [ ]:
```
df_test = df_train.applymap(lambda x: x.lower() if isinstance(x, str) else x)
df_test
```

```
C:\Users\sa\AppData\Local\Temp\ipykernel_10936\3176910913.py:1: FutureWarning: DataFrame.
applymap has been deprecated. Use DataFrame.map instead.
  df_test = df_train.applymap(lambda x: x.lower() if isinstance(x, str) else x)
```

Out[ ]:

	id	title	author	text	label
0	0	house dem aide: we didn't even see comey's let...	darrell lucus	house dem aide: we didn't even see comey's let...	1
1	1	flynn: hillary clinton, big woman on campus - ...	daniel j. flynn	ever get the feeling your life circles the rou...	0
2	2	why the truth might get you fired	consortiumnews.com	why the truth might get you fired october 29, ...	1
3	3	15 civilians killed in single us airstrike hav...	jessica purkiss	videos 15 civilians killed in single us airstr...	1
4	4	iranian woman jailed for fictional unpublished...	howard portnoy	print \nan iranian woman has been sentenced to...	1
...	...	...	...	...	...
20795	20795	rapper t.i.: trump a 'poster child for white s...	jerome hudson	rapper t. i. unloaded on black celebrities who...	0
20796	20796	n.f.l. playoffs: schedule, matchups and odds -...	benjamin hoffman	when the green bay packers lost to the washing...	0
20797	20797	macy's is said to receive takeover approach by...	michael j. de la merced and rachel abrams	the macy's of today grew from the union of sev...	0
20798	20798	nato, russia to hold parallel exercises in bal...	alex ansary	nato, russia to hold parallel exercises in bal...	1
20799	20799	what keeps the f-35 alive	david swanson	david swanson is an author, activist, journa...	1

18285 rows × 5 columns

Step 2: Punctuation

In [ ]:

```
import string

def remove_punctuation(text):
    no_punct=[words for words in text if words not in string.punctuation ]
    words_wo_punct=''.join(no_punct)
    return words_wo_punct

# Remove punctuation from both train and test dataset
df_train['title_wo_punct']=df_train['title'].apply(lambda x: remove_punctuation(x))
df_test['title_wo_punct']=df_test['title'].apply(lambda x: remove_punctuation(x))

df_train.head()
```

Out[ ]:

	id	title	author	text	label	title_wo_punct
0	0	house dem aide: we didn't even see comey's let...	darrell lucus	house dem aide: we didn't even see comey's let...	1	house dem aide we didn't even see comey's lett...
1	1	flynn: hillary clinton, big woman on campus - ...	daniel j. flynn	ever get the feeling your life circles the rou...	0	flynn hillary clinton big woman on campus bre...
2	2	why the truth might get you fired	consortiumnews.com	why the truth might get you fired october 29, ...	1	why the truth might get you fired
3	3	15 civilians killed in single us airstrike hav...	jessica purkiss	videos 15 civilians killed in single us airstr...	1	15 civilians killed in single us airstrike hav...
4	4	iranian woman jailed for fictional unpublished...	howard portnoy	print \nan iranian woman has been sentenced to...	1	iranian woman jailed for fictional unpublished...

Step 3: HTML Code , URL Links & emoji

In [ ]:
```
import re  # Import the re module for regular expressions

def text_clean(text):
    text = re.sub(r'^https?:\/\/.*[\r\n]*', '', text, flags=re.MULTILINE)
    text = re.sub('<.*?>+', '', text)
    regrex_pattern = re.compile(pattern = "["
        u"\U0001F600-\U0001F64F"  # emoticons
        u"\U0001F300-\U0001F5FF"  # symbols & pictographs
        u"\U0001F680-\U0001F6FF"  # transport & map symbols
        u"\U0001F1E0-\U0001F1FF"  # flags (iOS)
                           "]+", flags = re.UNICODE)
    text = regrex_pattern.sub(r'',text)
    text = ''.join([i for i in text if not i.isdigit()])
    return text
df_train['title_wo_punct_clean']=df_train['title_wo_punct'].apply(lambda x: text_clean(x))
df_test['title_wo_punct_clean']=df_test['title_wo_punct'].apply(lambda x: text_clean(x))
df_train.head()
```

Out[ ]:

	id	title	author	text	label	title_wo_punct	title_wo_punct_clean
0	0	house dem aide: we didn't even see comey's let...	darrell lucus	house dem aide: we didn't even see comey's let...	1	house dem aide we didn't even see comey's lett...	house dem aide we didn't even see comey's lett...
1	1	flynn: hillary clinton, big woman on campus - ...	daniel j. flynn	ever get the feeling your life circles the rou...	0	flynn hillary clinton big woman on campus bre...	flynn hillary clinton big woman on campus bre...
2	2	why the truth might get you fired	consortiumnews.com	why the truth might get you fired october 29, ...	1	why the truth might get you fired	why the truth might get you fired
3	3	15 civilians killed in single us airstrike hav...	jessica purkiss	videos 15 civilians killed in single us airstr...	1	15 civilians killed in single us airstrike hav...	civilians killed in single us airstrike have ...
4	4	iranian woman jailed for fictional unpublished...	howard portnoy	print \nan iranian woman has been sentenced to...	1	iranian woman jailed for fictional unpublished...	iranian woman jailed for fictional unpublished...

Step 4: Spell Checks

In [ ]:
```
from textblob import TextBlob

def correct_spelling(text):
    return str(TextBlob(text).correct())

# df_train['text_wo_punct_clean_spell'] = df_train['text_wo_punct_clean'].apply(correct_s
# df_train['title_wo_punct_clean_spell'] = df_train['title_wo_punct_clean'].apply(correct
# df_test['text_wo_punct_clean_spell'] = df_test['text_wo_punct_clean'].apply(correct_spe
# df_test['title_wo_punct_clean_spell'] = df_test['title_wo_punct_clean'].apply(correct_s
```

Step 4: Tokenization

Tokenizing is the process of splitting strings into a list of words. We will make use of Regular Expressions or regex to do the splitting. Regex can be used to describe a search pattern.

In [ ]:
```
def tokenize(text):
    split=re.split("\W+",text)
    return split
```

```
df_train['title_wo_punct_clean_spell_split']=df_train['title_wo_punct_clean'].apply(lambd
df_test['title_wo_punct_clean_spell_split']=df_test['title_wo_punct_clean'].apply(lambda
df_test['title_wo_punct_clean_spell_split']
```

```
<>:2: SyntaxWarning: invalid escape sequence '\W'
<>:2: SyntaxWarning: invalid escape sequence '\W'
C:\Users\sa\AppData\Local\Temp\ipykernel_10936\2286958798.py:2: SyntaxWarning: invalid es
cape sequence '\W'
  split=re.split("\W+",text)
```

Out[ ]:
```
0        [house, dem, aide, we, didn, t, even, see, com...
1        [flynn, hillary, clinton, big, woman, on, camp...
2                [why, the, truth, might, get, you, fired]
3        [, civilians, killed, in, single, us, airstrik...
4        [iranian, woman, jailed, for, fictional, unpub...
                              ...
20795    [rapper, ti, trump, a, poster, child, for, whi...
20796    [nfl, playoffs, schedule, matchups, and, odds,...
20797    [macy, s, is, said, to, receive, takeover, app...
20798    [nato, russia, to, hold, parallel, exercises, ...
20799                          [what, keeps, the, f, alive]
Name: title_wo_punct_clean_spell_split, Length: 18285, dtype: object
```

Step 5: Stop words

Stop words are irrelevant words that won't help in identifying a text as real or fake. We will use "nltk" library for stop-words and some of the stop words

In [ ]:
```
import nltk
from nltk.corpus import stopwords

stopword = nltk.corpus.stopwords.words('english')
print('Stopwords are:',stopword)
```

```
Stopwords are: ['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "yo
u're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'hi
m', 'his', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'its
elf', 'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 't
his', 'that', "that'll", 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'bee
n', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'th
e', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for',
'with', 'about', 'against', 'between', 'into', 'through', 'during', 'before', 'after', 'a
bove', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under', 'a
gain', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all',
'any', 'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'no
t', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just',
'don', "don't", 'should', "should've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ai
n', 'aren', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn',
"hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'ma', 'mightn', "might
n't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "shouldn't",
'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"]
```

In [ ]:
```
def remove_stopwords(text):
    text=[word for word in text if word not in stopword]
    return text

df_train['title_wo_punct_clean_spell_split_stopwords']=df_train['title_wo_punct_clean_spe
df_test['title_wo_punct_clean_spell_split_stopwords']=df_test['title_wo_punct_clean_spell
df_train.head()
```

Out[ ]:

	id	title	author	text	label	title_wo_punct	title_wo_punct_clean	title_wo_p
0	0	house dem aide: we didn’t even see comey’s let...	darrell lucus	house dem aide: we didn’t even see comey’s let...	1	house dem aide we didn’t even see comey’s lett...	house dem aide we didn’t even see comey’s lett...	[house,
1	1	flynn: hillary clinton, big woman on campus - ...	daniel j. flynn	ever get the feeling your life circles the rou...	0	flynn hillary clinton big woman on campus bre...	flynn hillary clinton big woman on campus bre...	[fly
2	2	why the truth might get you fired	consortiumnews.com	why the truth might get you fired october 29, ...	1	why the truth might get you fired	why the truth might get you fired	[why, the,
3	3	15 civilians killed in single us airstrike hav...	jessica purkiss	videos 15 civilians killed in single us airstr...	1	15 civilians killed in single us airstrike hav...	civilians killed in single us airstrike have ...	[, civilia
4	4	iranian woman jailed for fictional unpublished...	howard portnoy	print \nan iranian woman has been sentenced to...	1	iranian woman jailed for fictional unpublished...	iranian woman jailed for fictional unpublished...	[iran

Step 6: Normalization

Normalization brings all the words under on the roof by adding stemming and lemmatization

Stemming There are many variations of words that do not bring any new information and create redundancy. Take "*He likes to walk*" and "*He likes walking,*" for example. Both have the same meaning, so the stemming function will remove the suffix and convert "*walking*" *to* "*walk.*" The example in this guide uses the PorterStemmer module to conduct the process. You can use the snowball module for different languages.

In [ ]:

```
from nltk.stem.porter import PorterStemmer
# Stemming: Taking the root of the word
def stemming_text(word_list):
    porter_stemmer = PorterStemmer()
    stem_output = ' '.join([PorterStemmer().stem(word) for word in word_list])
    return stem_output
df_train['title_wo_punct_clean_spell_split_stopwords_stemp']=df_train['title_wo_punct_cle
df_test ['title_wo_punct_clean_spell_split_stopwords_stemp']=df_test['title_wo_punct_clea
df_train.head()
```

Out[ ]:

	id	title	author	text	label	title_wo_punct	title_wo_punct_clean	title_wo_p
0	0	house dem aide: we didn't even see comey's let...	darrell lucus	house dem aide: we didn't even see comey's let...	1	house dem aide we didn't even see comey's lett...	house dem aide we didn't even see comey's lett...	[house,
1	1	flynn: hillary clinton, big woman on campus - ...	daniel j. flynn	ever get the feeling your life circles the rou...	0	flynn hillary clinton big woman on campus bre...	flynn hillary clinton big woman on campus bre...	[fly
2	2	why the truth might get you fired	consortiumnews.com	why the truth might get you fired october 29, ...	1	why the truth might get you fired	why the truth might get you fired	[why, the,
3	3	15 civilians killed in single us airstrike hav...	jessica purkiss	videos 15 civilians killed in single us airstr...	1	15 civilians killed in single us airstrike hav...	civilians killed in single us airstrike have ...	[, civilia
4	4	iranian woman jailed for fictional unpublished...	howard portnoy	print \nan iranian woman has been sentenced to...	1	iranian woman jailed for fictional unpublished...	iranian woman jailed for fictional unpublished...	[iran

Lemmatization

Unlike stemming, *lemmatization* performs normalization using vocabulary and morphological analysis of words. *Lemmatization* aims to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma. *Lemmatization* uses a dictionary, which makes it slower than stemming, however the results make much more sense than what you get from stemming. *Lemmatization* is built on WordNet's built-in morphy function, making it an intelligent operation for text analysis. A WordNet module is a large and public lexical database for the English language. Its aim is to maintain the structured relationship between the words. The *WordNetLemmitizer()* is the earliest and most widely used function.

In [ ]:

```
from nltk.stem import WordNetLemmatizer
def lemmatize_text(word_list):
    lemmatizer = WordNetLemmatizer()
    lemmatized_output = ' '.join([lemmatizer.lemmatize(w) for w in word_list])
    return lemmatized_output

df_train['clean_title']=df_train['title_wo_punct_clean_spell_split_stopwords_stemp'].appl
df_test['clean_title']=df_test['title_wo_punct_clean_spell_split_stopwords_stemp'].apply(
df_test.head()
```

Out[ ]:

	id	title	author	text	label	title_wo_punct	title_wo_punct_clean	title_wo_p
0	0	house dem aide: we didn't even see comey's let...	darrell lucus	house dem aide: we didn't even see comey's let...	1	house dem aide we didn't even see comey's lett...	house dem aide we didn't even see comey's lett...	[house,
1	1	flynn: hillary clinton, big woman on campus - ...	daniel j. flynn	ever get the feeling your life circles the rou...	0	flynn hillary clinton big woman on campus bre...	flynn hillary clinton big woman on campus bre...	[fly
2	2	why the truth might get you fired	consortiumnews.com	why the truth might get you fired october 29, ...	1	why the truth might get you fired	why the truth might get you fired	[why, the,
3	3	15 civilians killed in single us airstrike hav...	jessica purkiss	videos 15 civilians killed in single us airstr...	1	15 civilians killed in single us airstrike hav...	civilians killed in single us airstrike have ...	[, civilia
4	4	iranian woman jailed for fictional unpublished...	howard portnoy	print \nan iranian woman has been sentenced to...	1	iranian woman jailed for fictional unpublished...	iranian woman jailed for fictional unpublished...	[iran

Embeddings

Word Embeddings or Word vectorization is a methodology in NLP to map words or phrases from vocabulary to a corresponding vector of real numbers To convert string data into numerical data one can use following methods

- One hot
- Bag of words
- TFIDF
- Word2Vec

In [ ]:
```
corpus_train = df_train['clean_title']
corpus_test = df_test['clean_title']
corpus_train[1]
```

Out[ ]:
```
'flynn hillary clinton big woman campus breitbart'
```

In [ ]:
```
from tensorflow.keras.preprocessing.text import one_hot
vocab_size = 10000
onehot_repr_test=[one_hot(words,vocab_size)for words in corpus_test]
onehot_repr_train=[one_hot(words,vocab_size)for words in corpus_train]
onehot_repr_train[1]
```

Out[ ]:
```
[5518, 3456, 5776, 3129, 8470, 4476, 7109]
```

In [ ]:
```
from sklearn.model_selection import train_test_split

#Split the CountVector vectorized data into train and test datasets for model training an
X_train, X_test, y_train, y_test =train_test_split(onehot_repr_train,df_train.label,test_
```

Padding

In [ ]:
```
from tensorflow.keras.preprocessing.sequence import pad_sequences

# Determine the maximum length of sequences (you can set it to any value based on your da
max_length = max(len(seq) for seq in onehot_repr_train)

# Pad sequences to ensure all sequences have the same length
X_train_padded = pad_sequences(X_train, maxlen=max_length, padding='post')
X_test_padded = pad_sequences(X_test, maxlen=max_length, padding='post')

# Convert to numpy arrays
X_train = np.array(X_train_padded)
X_test = np.array(X_test_padded)

# Now you can access the shape attribute
print('Shape of train:', X_train.shape)
print("Shape of validation:", X_test.shape)
```

```
Shape of train: (14628, 46)
Shape of validation: (3657, 46)
```

Creating Model

In [ ]:
```
import numpy as np
from tensorflow.keras.layers import LSTM
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense, Dropout

embedding_vector_features=40 ##features representation

# Build the model
model = Sequential()
# Add Embedding layer
model.add(Embedding(input_dim=vocab_size, output_dim=embedded_vector_features, input_leng
# Add Dropout layer
model.add(Dropout(0.3))
# Add LSTM layer
model.add(LSTM(100))
# Add Dropout layer
model.add(Dropout(0.3))
# Add Dense layer
model.add(Dense(1, activation='sigmoid'))
# Compile the model
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

```
c:\Users\sa\AppData\Local\Programs\Python\Python312\Lib\site-packages\keras\src\layers\co
re\embedding.py:90: UserWarning: Argument `input_length` is deprecated. Just remove it.
  warnings.warn(
```

Model Prediction

In [ ]:
```
model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=10, batch_size=64)
```

```
Epoch 1/10
229/229 ━━━━━━━━━━━━━━━━━━━━ 8s 30ms/step - accuracy: 0.5755 - loss: 0.6704 - val_accurac
y: 0.8305 - val_loss: 0.4598
Epoch 2/10
229/229 ━━━━━━━━━━━━━━━━━━━━ 6s 25ms/step - accuracy: 0.7697 - loss: 0.4639 - val_accurac
y: 0.7684 - val_loss: 0.4352
Epoch 3/10
229/229 ━━━━━━━━━━━━━━━━━━━━ 8s 33ms/step - accuracy: 0.8014 - loss: 0.4014 - val_accurac
y: 0.9007 - val_loss: 0.2497
Epoch 4/10
229/229 ━━━━━━━━━━━━━━━━━━━━ 8s 36ms/step - accuracy: 0.9243 - loss: 0.1909 - val_accurac
y: 0.9259 - val_loss: 0.1939
Epoch 5/10
229/229 ━━━━━━━━━━━━━━━━━━━━ 9s 41ms/step - accuracy: 0.9649 - loss: 0.1133 - val_accurac
y: 0.9300 - val_loss: 0.1887
Epoch 6/10
229/229 ━━━━━━━━━━━━━━━━━━━━ 8s 34ms/step - accuracy: 0.9777 - loss: 0.0858 - val_accurac
y: 0.9295 - val_loss: 0.2075
Epoch 7/10
229/229 ━━━━━━━━━━━━━━━━━━━━ 9s 39ms/step - accuracy: 0.9817 - loss: 0.0697 - val_accurac
y: 0.9308 - val_loss: 0.2309
Epoch 8/10
229/229 ━━━━━━━━━━━━━━━━━━━━ 9s 41ms/step - accuracy: 0.9886 - loss: 0.0512 - val_accurac
y: 0.9259 - val_loss: 0.2621
Epoch 9/10
229/229 ━━━━━━━━━━━━━━━━━━━━ 11s 47ms/step - accuracy: 0.9898 - loss: 0.0494 - val_accura
cy: 0.9251 - val_loss: 0.2613
Epoch 10/10
229/229 ━━━━━━━━━━━━━━━━━━━━ 10s 43ms/step - accuracy: 0.9907 - loss: 0.0435 - val_accura
cy: 0.9262 - val_loss: 0.2568
```

Out[ ]:
```
<keras.src.callbacks.history.History at 0x21e7b73de80>
```

In [ ]:
```
print(model.summary())
```

Model: "sequential_4"

Layer (type)	Output Shape	Param #
embedding_3 (Embedding)	(None, 46, 40)	400,000
dropout_6 (Dropout)	(None, 46, 40)	0
lstm_3 (LSTM)	(None, 100)	56,400
dropout_7 (Dropout)	(None, 100)	0
dense_3 (Dense)	(None, 1)	101

Total params: 1,369,505 (5.22 MB)

Trainable params: 456,501 (1.74 MB)

Non-trainable params: 0 (0.00 B)

Optimizer params: 913,004 (3.48 MB)

None

Performance Metrics

In [ ]:
```
y_pred = model.predict(X_test)
```

```
115/115 ━━━━━━━━━━━━━━━━━━━━ 2s 13ms/step
```

In [ ]:
```
# AUC ROC Curve
y_pred = np.where(y_pred>0.5, 1, 0)
y_pred
```

Out[ ]:
```
array([[1],
       [1],
       [0],
       ...,
       [0],
       [0],
       [0]])
```

In [ ]:
```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test, y_pred)
```

Out[ ]:
```
array([[1945,  157],
       [ 113, 1442]], dtype=int64)
```

In [ ]:
```
from sklearn.metrics import accuracy_score
accuracy_score(y_test, y_pred)
```

Out[ ]:
```
0.92616899097621
```

In [ ]:
```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.95      0.93      0.94      2102
           1       0.90      0.93      0.91      1555

    accuracy                           0.93      3657
   macro avg       0.92      0.93      0.92      3657
weighted avg       0.93      0.93      0.93      3657
```