In [3]:

```python
 import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
data = pd.read_csv("fake_news.csv")
data.head()
data.shape
data.info()
data.isna().sum()
data = data.drop(['id'] , axis=1)
 data = data.fillna('')
data['content'] = data['author'] +''+ data['title']+''+data['text']
 data = data.drop(['title','author','text'], axis=1)
data.head()
```

```
<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
```

Out[3]:

	label	content
0	1	Darrell LucusHouse Dem Aide: We Didn’t Even Se...
1	0	Daniel J. FlynnFLYNN: Hillary Clinton, Big Wom...
2	1	Consortiumnews.comWhy the Truth Might Get You ...
3	1	Jessica Purkiss15 Civilians Killed In Single U...
4	1	Howard PortnoyIranian woman jailed for fiction...

In [4]:

```python
data['content'] = data['content'].apply(lambda x: " ".join(x.lower() for x in x.split()))
```

In [5]:

```python
data['content'] = data['content'].str.replace('[^\w\s]','')
```

```
C:\Users\Sai Krishna Hari\AppData\Local\Temp\ipykernel_5984\3643324700.py:1: FutureWarning: The default valu
e of regex will change from True to False in a future version.
  data['content'] = data['content'].str.replace('[^\w\s]','')
```

In [6]:

```python
import nltk
nltk.download("stopwords")
```

```
[nltk_data] Downloading package stopwords to C:\Users\Sai Krishna
[nltk_data]     Hari\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[6]:

```
True
```

In [8]:

```python
from nltk.corpus import stopwords
stop=stopwords.words('english')
data['content']=data['content'].apply(lambda x:" ".join(x for x in x.split() if x not in stop))
```

In [11]:

```python
from nltk.stem import WordNetLemmatizer
from textblob import Word
data['content']=data['content'].apply(lambda x:"".join([Word(word).lemmatize() for word in x.split()]))
data['content'].head()
```

Out[11]:

```
0    darrelllucushousedemaidedidntevenseecomeyslett...
1    danieljflynnflynnhillaryclintonbigwomancampusb...
2    consortiumnewscomwhytruthmightgetfiredwhytruth...
3    jessicapurkiss15civiliankilledsingleuairstrike...
4    howardportnoyiranianwomanjailedfictionalunpubl...
Name: content, dtype: object
```

In [12]:

```python
x=data[['content']]
y=data['label']
```

In [13]:

```
from sklearn.model_selection import train_test_split
```

In [15]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=45,stratify=y)
```

In [16]:

```
print(x_train.shape)
```

```
(20755, 1)
```

In [17]:

```
print(x_test.shape)
```

```
(45, 1)
```

In [18]:

```
print(y_train.shape)
```

```
(20755,)
```

In [19]:

```
print(y_test.shape)
```

```
(45,)
```

In [21]:

```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf_vect = TfidfVectorizer(analyzer='word', token_pattern=r'\w{1,}', max_features=5000)
tfidf_vect.fit(data['content'])
xtrain_tfidf = tfidf_vect.transform(x_train['content'])
xtest_tfidf = tfidf_vect.transform(x_test['content'])

```

In [ ]:

```

```