In [1]:
```
import pandas as pd
import nltk
import re

from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from sklearn.preprocessing import LabelEncoder
from sklearn.feature_extraction.text import TfidfVectorizer

nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('omw-1.4')
```

```
[nltk_data] Downloading package punkt to C:\Users\Lalit
[nltk_data]     H\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to C:\Users\Lalit
[nltk_data]     H\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to C:\Users\Lalit
[nltk_data]     H\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package omw-1.4 to C:\Users\Lalit
[nltk_data]     H\AppData\Roaming\nltk_data...
[nltk_data]   Package omw-1.4 is already up-to-date!
```

Out[1]: True

In [2]:
```
df = pd.read_pickle("News_dataset.pickle")

df.head()
```

Out[2]:

	File_Name	Content	Category	Complete_Filename	id	News_length
0	001.txt	Ad sales boost Time Warner profit\r\n\r\nQuart...	business	001.txt-business	1	2569
1	002.txt	Dollar gains on Greenspan speech\r\n\r\nThe do...	business	002.txt-business	1	2257
2	003.txt	Yukos unit buyer faces loan claim\r\n\r\nThe o...	business	003.txt-business	1	1557
3	004.txt	High fuel prices hit BA's profits\r\n\r\nBriti...	business	004.txt-business	1	2421
4	005.txt	Pernod takeover talk lifts Domecq\r\n\r\nShare...	business	005.txt-business	1	1575

In [3]:
```
print(df.columns)
```

```
Index(['File_Name', 'Content', 'Category', 'Complete_Filename', 'id',
       'News_length'],
      dtype='object')
```

In [7]:
```
def clean_text(text):
    text = text.lower()
    text = re.sub(r'[^a-z\s]', '', text)
    return text
```

```
df['clean_text'] = df['Content'].apply(clean_text)

df[['Content', 'clean_text']].head()
```

Out[7]:

	Content	clean_text
0	Ad sales boost Time Warner profit\r\n\r\nQuart...	ad sales boost time warner profit\r\n\r\nquart...
1	Dollar gains on Greenspan speech\r\n\r\nThe do...	dollar gains on greenspan speech\r\n\r\nthe do...
2	Yukos unit buyer faces loan claim\r\n\r\nThe o...	yukos unit buyer faces loan claim\r\n\r\nthe o...
3	High fuel prices hit BA's profits\r\n\r\nBriti...	high fuel prices hit bas profits\r\n\r\nbritis...
4	Pernod takeover talk lifts Domecq\r\n\r\nShare...	pernod takeover talk lifts domecq\r\n\r\nshare...

In [8]:

```
stop_words = set(stopwords.words('english'))
lemmatizer = WordNetLemmatizer()

def preprocess(text):
    tokens = nltk.word_tokenize(text)
    tokens = [word for word in tokens if word not in stop_words]
    tokens = [lemmatizer.lemmatize(word) for word in tokens]
    return tokens

df['tokens'] = df['clean_text'].apply(preprocess)

df[['clean_text', 'tokens']].head()
```

Out[8]:

	clean_text	tokens
0	ad sales boost time warner profit\r\n\r\nquart...	[ad, sale, boost, time, warner, profit, quarte...
1	dollar gains on greenspan speech\r\n\r\nthe do...	[dollar, gain, greenspan, speech, dollar, hit,...
2	yukos unit buyer faces loan claim\r\n\r\nthe o...	[yukos, unit, buyer, face, loan, claim, owner,...
3	high fuel prices hit bas profits\r\n\r\nbritis...	[high, fuel, price, hit, ba, profit, british, ...
4	pernod takeover talk lifts domecq\r\n\r\nshare...	[pernod, takeover, talk, lift, domecq, share, ...

In [9]:

```
df['final_text'] = df['tokens'].apply(lambda x: ' '.join(x))
```

In [10]:

```
label_encoder = LabelEncoder()

df['encoded_label'] = label_encoder.fit_transform(df['Category'])

print(df[['Category', 'encoded_label']])
```

```
      Category  encoded_label
0     business              0
1     business              0
2     business              0
3     business              0
4     business              0
...        ...            ...
2220      tech              4
2221      tech              4
2222      tech              4
2223      tech              4
2224      tech              4

[2225 rows x 2 columns]
```

In [11]:
```
tfidf_vectorizer = TfidfVectorizer()

tfidf_matrix = tfidf_vectorizer.fit_transform(df['final_text'])

print("TF-IDF Shape:", tfidf_matrix.shape)
```

```
TF-IDF Shape: (2225, 27879)
```

In [12]:
```
df.to_csv("processed_news.csv", index=False)

tfidf_df = pd.DataFrame(
    tfidf_matrix.toarray(),
    columns=tfidf_vectorizer.get_feature_names_out()
)

tfidf_df.to_csv("tfidf_output.csv", index=False)

print("Files saved successfully!")
```

```
Files saved successfully!
```

In [ ]:

In [ ]: