Name: Onasvee Banarse

Class: BE Computer-1

Roll No: 09

Problem Statement: Perform text cleaning, perform lemmatization (any method), remove stop words (any method), label encoding. Create representations using TF-IDF. Save outputs. Dataset: https://github.com/PICT-NLP/BE-NLP-Elective/blob/main/3-Preprocessing/News_dataset.pickle

In [1]:

```
import pandas as pd
import pickle
import re
from nltk import WordNetLemmatizer, word_tokenize
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.preprocessing import LabelEncoder
```

In [2]:

```
# Load the dataset
with open('News_dataset.pickle', 'rb') as f:
    news = pickle.load(f)
```

In [3]:

```
news
```

Out[3]:

	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
...	...	...	...	...	...	...
2220	397.txt	BT program to beat dialler scams\r\n\r\nBT is ...	tech	397.txt-tech	1	2526
2221	398.txt	Spam e-mails tempt net shoppers\r\n\r\nCompute...	tech	398.txt-tech	1	2294
2222	399.txt	Be careful how you code\r\n\r\nA new European ...	tech	399.txt-tech	1	6297
2223	400.txt	US cyber security chief resigns\r\n\r\nThe man...	tech	400.txt-tech	1	2323
2224	401.txt	Losing yourself in online gaming\r\n\r\nOnline...	tech	401.txt-tech	1	16248

2225 rows × 6 columns

In [4]:

```
df = pd.DataFrame(news, columns=['Content', 'Category'])
```

In [5]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2225 entries, 0 to 2224
Data columns (total 2 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   Content   2225 non-null   object
 1   Category  2225 non-null   object
dtypes: object(2)
memory usage: 34.9+ KB
```

Text cleaning, Lemmatization and Stop word removal

In [6]:

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

# define a function for text cleaning, lemmatization and stop word removal
def clean_text(text):
    text = re.sub(r'[^\w\s]', '', text) # remove punctuation
    text = text.lower() # convert to lowercase
    tokens = word_tokenize(text) # tokenize the text
    tokens = [lemmatizer.lemmatize(token) for token in tokens] # lemmatize the toke
    tokens = [token for token in tokens if token not in stop_words] # remove stop w
    clean_text = ' '.join(tokens)
    return clean_text

# apply the function to the 'news' column
df['clean_text'] = df['Content'].apply(clean_text)
```

Label encoding

In [12]:

```
# label encode the 'category' column
le = LabelEncoder()
df['Category'] = le.fit_transform(df['Category'])
df.head()
```

Out[12]:

	Content	Category	clean_text
0	Ad sales boost Time Warner profit\r\n\r\nQuart...	0	ad sale boost time warner profit quarterly pro...
1	Dollar gains on Greenspan speech\r\n\r\nThe do...	0	dollar gain greenspan speech dollar ha hit hig...
2	Yukos unit buyer faces loan claim\r\n\r\nThe o...	0	yukos unit buyer face loan claim owner embattl...
3	High fuel prices hit BA's profits\r\n\r\nBriti...	0	high fuel price hit ba profit british airway h...
4	Pernod takeover talk lifts Domecq\r\n\r\nShare...	0	pernod takeover talk lift domecq share uk drin...

TF-IDF

In [9]:

```
# create TF-IDF representations of the clean text
tfidf_vec = TfidfVectorizer()
tfidf_count_occurs = tfidf_vec.fit_transform(df['clean_text'])
tfidf_count_occur_df = pd.DataFrame((count, word) for word, count in zip(
tfidf_count_occurs.toarray().tolist()[0], tfidf_vec.get_feature_names()))
tfidf_count_occur_df.columns = ['Word', 'Count']
tfidf_count_occur_df.sort_values('Count', ascending=False, inplace=True)
tfidf_count_occur_df.head()
```

Out[9]:

	Word	Count
27401	timewarner	0.487146
21674	profit	0.344867
3442	aol	0.257683
29256	warner	0.210784
23199	revenue	0.141471

Save Outputs

In [10]:

```
# save the processed data and the TF-IDF vectorizer
with open('processed_data.pickle', 'wb') as f:
    pickle.dump(df, f)
```

```python
with open('tfidf.pickle', 'wb') as f:
    pickle.dump(tfidf_vec, f)
```