# FAKE NEWS DETECTION USING NLP

Team Leader : Lokesh.T

Team Member-1 : Harish.C

Team Member-2 : Manikandan.S

Team Member-3 : Richard Aaron.S

---

**Phase3:** Loading and preprocessing the dataset.

**Introduction:**
In an era dominated by information, distinguishing between credible news and misleading information is more crucial than ever. The "Detecting Fake News Using Natural Language Processing" project aims to tackle this challenge by harnessing the power of NLP

**Source:**
The dataset is obtained from kaggle.com.
Dataset Link:
https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset

**Loading and Preprocessing:**
**Loading the dataset:**
Load the dataset by using pandas package library.
**Syntax:** variable_name=pd.read_csv('Dataset Path')

```
# Importing the necessary libraries

import pandas as pd
import nltk
import re
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
from sklearn.feature_extraction.text import TfidfVectorizer

# Download necessary NLTK resources

nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')

# Loading fake dataset

fake_data=pd.read_csv('/content/Fake.csv')
print(fake_data.info())
print(fake_data.head())
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23481 entries, 0 to 23480
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   title    23481 non-null  object
 1   text     23481 non-null  object
 2   subject  23481 non-null  object
 3   date     23481 non-null  object
dtypes: object(4)
memory usage: 733.9+ KB
None
                                               title  \
0   Donald Trump Sends Out Embarrassing New Year’...
1   Drunk Bragging Trump Staffer Started Russian ...
2   Sheriff David Clarke Becomes An Internet Joke...
3   Trump Is So Obsessed He Even Has Obama’s Name...
4   Pope Francis Just Called Out Donald Trump Dur...

                                                text subject  \
0  Donald Trump just couldn t wish all Americans ...    News
1  House Intelligence Committee Chairman Devin Nu...    News
2  On Friday, it was revealed that former Milwauk...    News
3  On Christmas day, Donald Trump announced that ...    News
4  Pope Francis used his annual Christmas Day mes...    News

                date
0  December 31, 2017
1  December 31, 2017
2  December 30, 2017
3  December 29, 2017
4  December 25, 2017
```

```
# Loading true dataset

true_data = pd.read_csv('/content/True.csv')
print(true_data.info())
print(true_data.head())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21417 entries, 0 to 21416
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   title    21417 non-null  object
 1   text     21417 non-null  object
 2   subject  21417 non-null  object
 3   date     21417 non-null  object
dtypes: object(4)
memory usage: 669.4+ KB
None
                                               title  \
0  As U.S. budget fight looms, Republicans flip t...
1  U.S. military to accept transgender recruits o...
2  Senior U.S. Republican senator: 'Let Mr. Muell...
```

```
3  FBI Russia probe helped by Australian diplomat...
4  Trump wants Postal Service to charge 'much mor...

                                              text       subject  \
0  WASHINGTON (Reuters) - The head of a conservat...  politicsNews
1  WASHINGTON (Reuters) - Transgender people will...  politicsNews
2  WASHINGTON (Reuters) - The special counsel inv...  politicsNews
3  WASHINGTON (Reuters) - Trump campaign adviser ...  politicsNews
4  SEATTLE/WASHINGTON (Reuters) - President Donal...  politicsNews

                 date
0  December 31, 2017
1  December 29, 2017
2  December 31, 2017
3  December 30, 2017
4  December 29, 2017
```

```
# Combine the datasets into one

data = pd.concat([fake_data, true_data], ignore_index=True)
print(data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 44898 entries, 0 to 44897
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   title    44898 non-null  object
 1   text     44898 non-null  object
 2   subject  44898 non-null  object
 3   date     44898 non-null  object
dtypes: object(4)
memory usage: 1.4+ MB
None
```

```
# Data preprocess

# Data Cleaning with regular expressions

data['text'] = data['text'].apply(lambda x: re.sub('<[^>]+>', '', x))  # Remove HTML tags
data['text'] = data['text'].apply(lambda x: re.sub('[^a-zA-Z\s]', '', x))  # Remove non-alphabetical characters

# Convert text to lowercase

data['text'] = data['text'].str.lower()

# Tokenization

data['tokens'] = data['text'].apply(word_tokenize)

# Stopword Removal

stop_words = set(stopwords.words('english'))
data['filtered_tokens'] = data['tokens'].apply(lambda tokens: [word for word in tokens if word not in stop_words])

# Text Lemmatization

lemmatizer = WordNetLemmatizer()
data['lemmatized_tokens'] = data['filtered_tokens'].apply(lambda tokens: [lemmatizer.lemmatize(word) for word in tokens])

# Text Vectorization (using TF-IDF)

tfidf_vectorizer = TfidfVectorizer(max_features=1000)  # Adjust max_features as needed
X_tfidf = tfidf_vectorizer.fit_transform(data['lemmatized_tokens'].apply(' '.join))
print(X_tfidf)

# Preprocessed data is stored in 'X_tfidf'
```

```
  (0, 412)      0.032011987464638327
  (0, 544)      0.04715775560746116
  (0, 995)      0.046959889230305786
  (0, 992)      0.050990578562527956
  (0, 669)      0.03904010224888849
  (0, 371)      0.041400420511736556
  (0, 140)      0.07802198535651134
  (0, 955)      0.05172226143553143
  (0, 615)      0.05077452290101154
  (0, 886)      0.04443692514037917
  (0, 477)      0.027537086353484452
  (0, 43)       0.048655394042109244
  (0, 986)      0.03505586289651313
  (0, 158)      0.049908297884010355
  (0, 546)      0.04654886397111031
  (0, 600)      0.042972532442220615
```

```
          (0, 518)      0.05030412359503157
          (0, 468)      0.06569857494682693
          (0, 516)      0.04820526360505604
          (0, 421)      0.032541581782343725
          (0, 780)      0.026675494595917688
          (0, 985)      0.04352164080359294
          (0, 385)      0.10170739235609522
          (0, 644)      0.04703396650413406
          (0, 528)      0.0614023954244021
          :     :
          (44897, 264)  0.14385634527687785
          (44897, 217)  0.11782634864302205
          (44897, 175)  0.10548431379144897
          (44897, 641)  0.11510072277256499
          (44897, 898)  0.11277185160818298
          (44897, 853)  0.08384912792046674
          (44897, 332)  0.09851037146906864
          (44897, 196)  0.069044155052663
          (44897, 243)  0.12408661281426923
          (44897, 557)  0.17874451888513132
          (44897, 293)  0.13614226455525055
          (44897, 626)  0.10940856468339578
          (44897, 856)  0.08923685298810526
          (44897, 922)  0.2091338657258787
          (44897, 852)  0.11357843625178249
          (44897, 773)  0.12755795850699497
          (44897, 931)  0.1447964669880508
          (44897, 769)  0.10963065802949314
          (44897, 768)  0.31536755319503784
          (44897, 419)  0.12401185287405021
          (44897, 41)   0.060497164994166755
          (44897, 959)  0.0800154356920179
          (44897, 198)  0.14114901030057514
          (44897, 994)  0.06231196213227771
          (44897, 591)  0.06603776045951681
```