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

Problem Definition and Design Thinking

Problem Definition:

The problem is to develop a fake news detection model using a Kaggle dataset. The goal is to distinguish between genuine and fake news articles based on their titles and text. This project involves using natural language processing (NLP) techniques to preprocess the text data, building a machine learning model for classification, and evaluating the model's performance.

Detecting fake news using NLP involves developing a system that can analyze news articles to distinguish between reliable and unreliable information. The problem is to use linguistic patterns, sentiment analysis, and contextual clues to create algorithms that assess the authenticity of news sources and claims. By employing Natural Language Processing, the goal is to provide a tool that helps users, journalists, and fact-checkers identify misinformation, promoting accurate news dissemination in an era where the spread of fake news is a significant concern.

By addressing the fake news problem, this technology aims to enhance information integrity, empower users to make informed decisions, and preserve the trustworthiness of news sources in the digital landscape. It's a dynamic field that continually evolves to counter emerging tactics used by purveyors of misinformation.

We are going to analyse the datatset present in kaggle “Fake and real news dataset”.

We planned to implement algorithms to clean and preprocess the raw data. This may include handling missing data, outlier detection, and normalization.

Using statistical analysis we will find unknown trends, patterns, and anomalies in the data.

Design Thinking:

- **Data Source**: Identify an available dataset containing fake news and real news.
- **Data Preprocessing**: Clean, transform, and prepare the dataset for analysis.
- **Feature Extraction**: Extract relevant features and metrics from the fake and real news data.
- **Model Selection**: Choose a suitable NLP model for classification. Common choices include:
  - Logistic Regression
  - Naive Bayes
  - Support Vector Machines
  - Deep Learning models (e.g., LSTM, BERT)

- **Model training:** Train the selected model using the preprocessed and vectorized data. The model learns the patterns that distinguish fake from real news.

- **Evaluation**: Assess the model's performance on a separate test dataset to ensure it generalizes well to new, unseen data.

1.Data Source: We are going to consider the dataset available in kaggle website “Fake and real news dataset”.

Dataset link: https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset

2. Data Preprocessing: We need to identify and handle missing data, outliers, and duplicates and ensure correct data types and formats.

Generate summary statistics and visualizations.

3.Feature Extraction: We need to Identify relevant features that can provide insights to detect fake and real news.

It involves Linguistic Patterns,Sentiment Analysis,Named Entities,Source Credibility Analysis and Contextual Analysis.

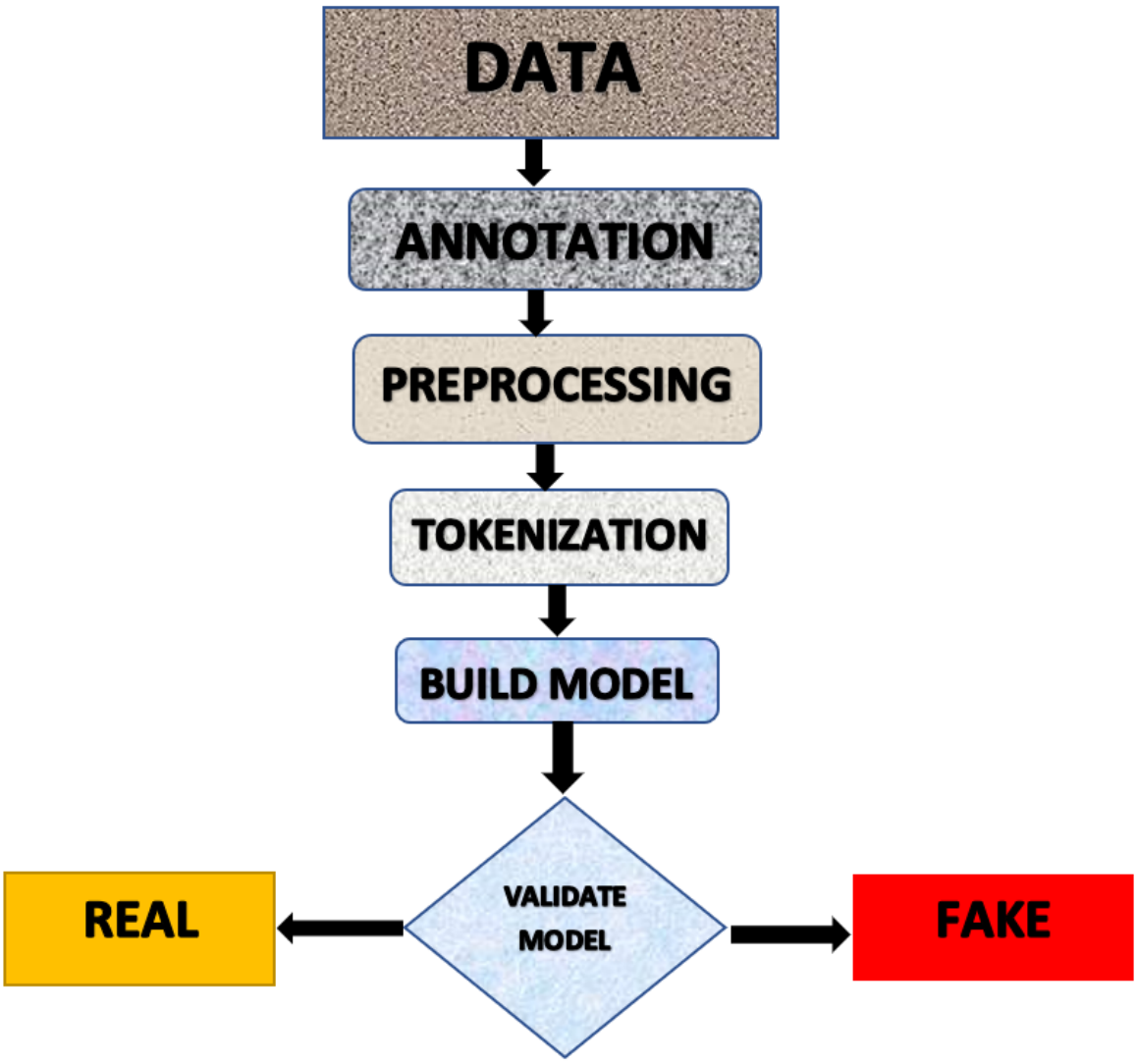
4.Model Selection: Select an NLP model (e.g., Naive Bayes, LSTM, BERT). Teach the model to recognize patterns in the data.

Test the model on new data to see how well it performs.

5. Model Training: Based on the available datasets the model is trained which is then used to predict the fake and real news for the user data.

6.Evaluation: The evaluation of data includes Testing of data,Predictions,Metrics,Confusion matrix and Real world testing.

Graphical representation:



Innovation:

- **Advanced NLP Models:**

Utilize state-of-the-art NLP models like transformers (e.g., BERT, GPT-3) to understand the context and semantics of news articles effectively. Implement fine-tuning and transfer learning techniques to adapt pretrained models to the specific task of fake news detection.

- **Contextual Understanding:**

Develop models that consider the broader context in which news articles are published, including historical events, political climate, and the reputation of the news source.

Incorporate temporal analysis to track the evolution of news stories over time, identifying changes in narratives or updates.

- **User Behavior Analysis:**

Analyze user interactions with news articles on social media platforms to identify patterns of information sharing and amplification of fake news.

- **Real-time Monitoring:**

Create a system that continuously monitors news sources and social media for the emergence of potentially fake news stories.

Utilize streaming data processing to keep the detection system up-to date in real-time.

- **User Feedback Loop:**

Implement a user feedback mechanism where users can report potentially fake news articles and provide feedback on system predictions.

Use user input to improve the accuracy and performance of the fake news detection system over time.

- **Web application integration:**

The Fake News Detection Web Application is designed to combat the spread of misinformation by providing users with a user-friendly platform to access and verify news articles' credibility. The application leverages Natural Language Processing (NLP) and user interaction to deliver accurate and up-to-date assessments of news articles.

Language used : PYTHON

Algorithm used : BERT

- BERT stands for Biredirectional Encoder Representations from Transfers.

- It is a machine learning(ML) framework for Natural Language Processing.

Required Modules:

- **pandas -** To analyse given dataset.

- **numpy -** To perform array operations.

- **matplotlib** - To plot various graphs.

- **tensorflow** - To implement data automation,model tracking.

- **transformers** - To provide an easy interface to use BERT with TensorFlow.

```
# 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
```

```
!pip install transformers

# 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
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from transformers import BertTokenizer, BertForSequenceClassification, AdamW
import torch

# 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())

# Loading true dataset
true_data = pd.read_csv('/content/True.csv')
print(true_data.info())
print(true_data.head())

# Combine the datasets into one
fake_data['label'] = 1  # Assigning label 1 to fake news
true_data['label'] = 0  # Assigning label 0 to true news
data = pd.concat([fake_data, true_data], ignore_index=True)
print(data.info())

# 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))

# Model Training and Evaluation

# Split the data
X_train, X_test, y_train, y_test = train_test_split(X_tfidf, data['label'], test_size=0.2, random_state=42)

# Naive Bayes
nb_model = MultinomialNB()
nb_model.fit(X_train, y_train)
nb_pred = nb_model.predict(X_test)
nb_accuracy = accuracy_score(y_test, nb_pred)


# Random Forest
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
rf_pred = rf_model.predict(X_test)
rf_accuracy = accuracy_score(y_test, rf_pred)

print(f'Naive Bayes Accuracy: {nb_accuracy}')
```

```
print(f'Random Forest Accuracy: {rf_accuracy}')
```

```
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
<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
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 44898 entries, 0 to 44897
Data columns (total 5 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
 4   label    44898 non-null  int64
dtypes: int64(1), object(4)
memory usage: 1.7+ MB
None
Naive Bayes Accuracy: 0.9200445434298441
Random Forest Accuracy: 0.9978841870824053
```