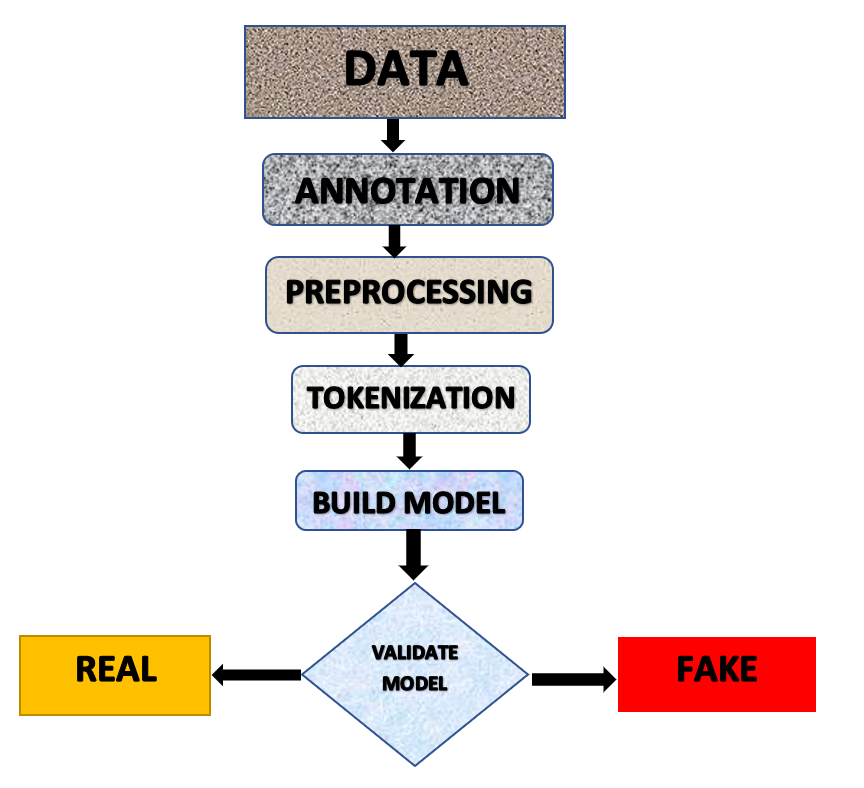
## **Fake News Detection Using NLP**

TEAM MEMBER

**311121205015 : BYLEEN JANET ROY J**

**Phase - V Document Submission**

**Project : Fake News Detection using NLP**



# 

# **Problem Statement:**

# The proliferation of false information through fake news in the contemporary digital environment has emerged as a substantial impediment, significantly impacting public conversations, eroding trust, and influencing decision-making processes. To address this prevalent issue, this initiative endeavors to devise a robust solution by crafting an efficient Fake News Detection Model using a Kaggle dataset. Employing advanced Natural Language Processing (NLP) methodologies, machine learning algorithms, and meticulous assessment techniques, the primary goal is to discern authentic news from fabricated articles based on their textual constructs. This endeavor delineates a structured methodology, commencing from dataset curation to model construction, with the overarching aim of mitigating the dissemination of misleading information.

# **Abstract:**

# The pervasive dissemination of fake news in today's digital landscape poses a critical challenge, influencing public discourse, trust, and decision-making. To tackle this pressing issue, this document presents a comprehensive solution for the development of an effective Fake News Detection Model using a Kaggle dataset. Leveraging Natural Language Processing (NLP) techniques, machine learning algorithms, and rigorous evaluation, our objective is to distinguish between genuine and fake news articles based on their textual content. This solution outlines a systematic approach from dataset selection to model development, aiming to combat the spread of misinformation.

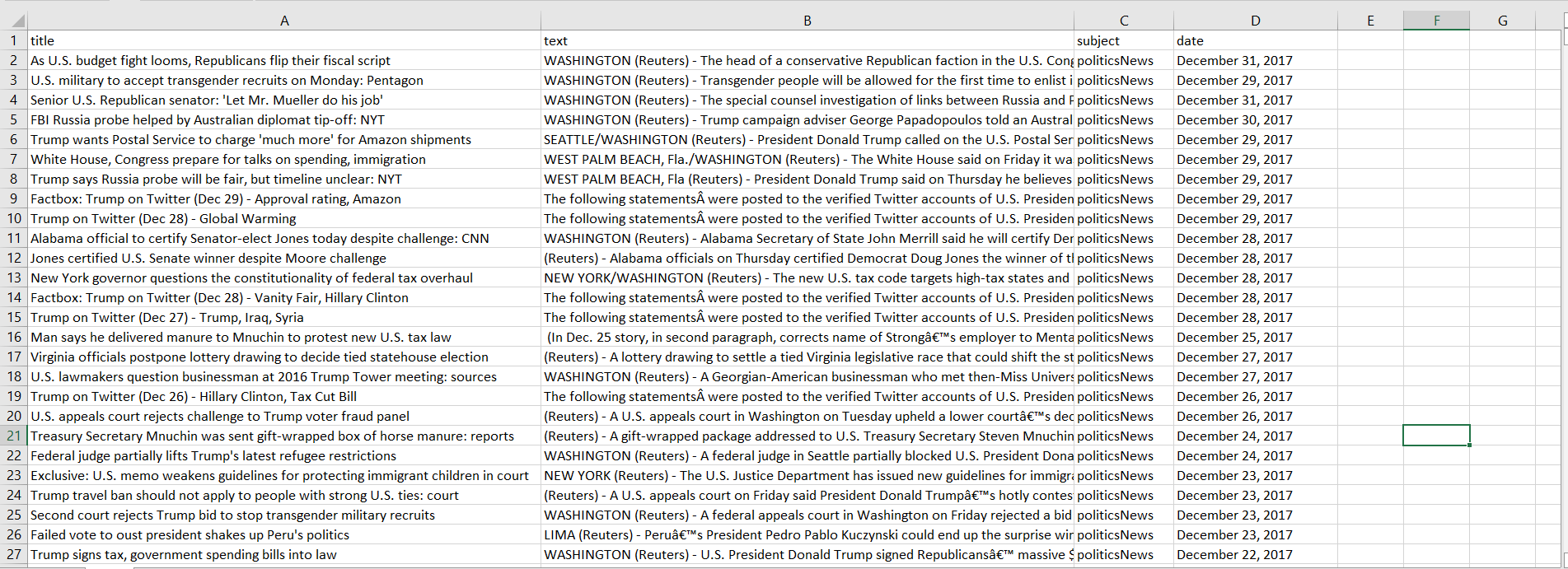
# **Introduction:**

# The rapid growth of the internet and social media has democratized the creation and distribution of news, but it has also given rise to a proliferation of fake news. Misleading or fabricated information can have far-reaching consequences, eroding trust in reliable news sources, influencing public opinion, and even impacting political processes. To address this challenge, we propose the development of a Fake News Detection Model, a critical tool in the fight against misinformation.

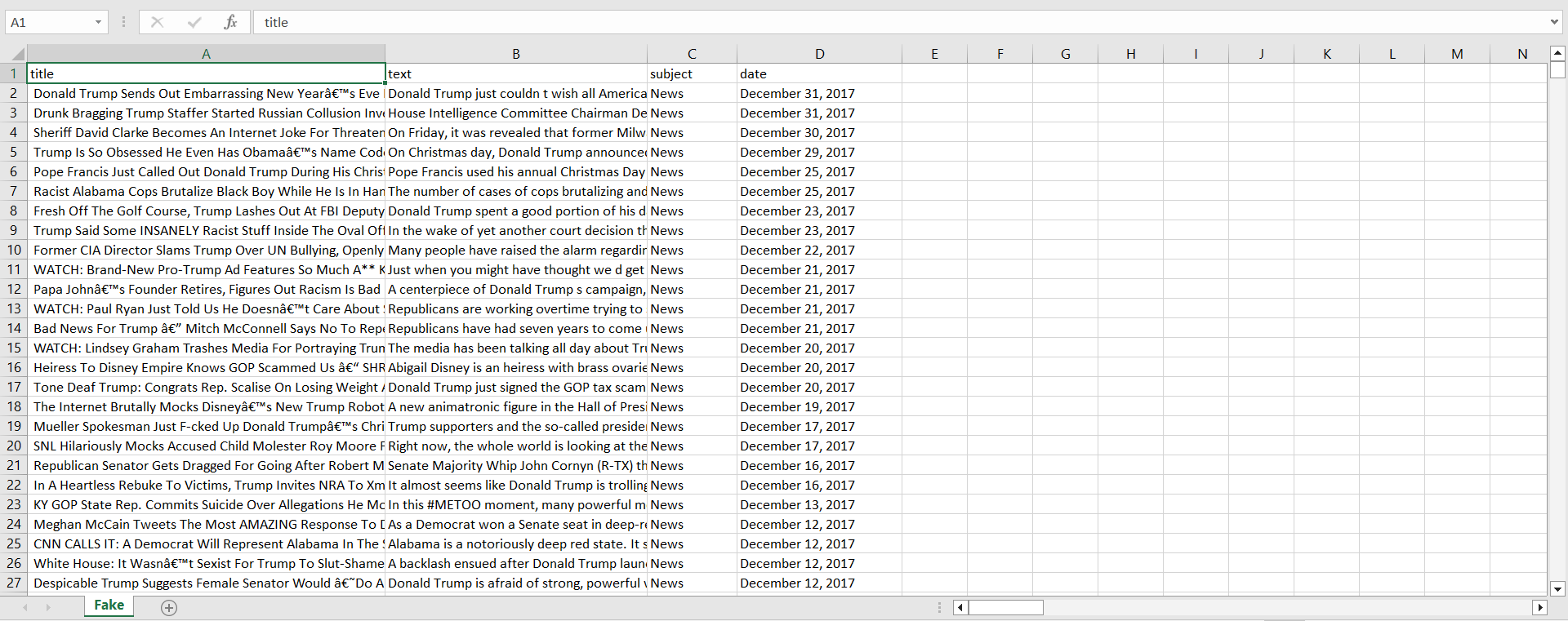
## **Dataset Source:**

Our journey to creating an effective Fake News Detection Model begins with the careful selection of a dataset. Kaggle, a reputable platform for data science, offers a diverse range of datasets, and for this project, we have chosen one that contains news articles' titles and text, along with labels indicating their authenticity (genuine or fake). This dataset serves as the foundation upon which we will construct our model.

**True.csv**

****

**Fake.csv**

****

In this section, we load the dataset from Kaggle.

The datasets are downloaded from kaggle and loaded.

The pd.read\_csv() function reads data from CSV files and stores it in Pandas DataFrames. The fake\_news and real\_news DataFrames will contain the fake and real news data, respectively.

## 

## 

## 

## **Data Preprocessing:**

### **Cleaning and Standardization:**

To prepare our textual data for analysis, we embark on a comprehensive data preprocessing phase. This phase encompasses several essential steps:

* Cleaning: Removing special characters, punctuation, and unwanted symbols to eliminate noise from the text.
* Tokenization: Breaking down text into individual words or tokens for analysis.
* Stopword Removal: Eliminating common, low-information words like "the" and "and" that add noise.
* Lowercasing: Converting all text to lowercase to ensure uniformity.
* Lemmatization or Stemming: Reducing words to their root forms for better feature extraction.

Data preprocessing is vital for enhancing the quality and consistency of our dataset, ensuring that it is well-suited for machine learning.

**PYTHON PROGRAM:**

import pandas as pd

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, f1\_score, roc\_auc\_score

from sklearn.linear\_model import LogisticRegression

from tensorflow.keras.preprocessing.text import Tokenizer

from tensorflow.keras.preprocessing.sequence import pad\_sequences

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Embedding, LSTM, Dense

# Load the "Fake.csv" dataset

fake\_data = pd.read\_csv("C:\\Users\\Bylee\\Downloads\\Fake.csv\\Fake.csv")

# Load the "True.csv" dataset

true\_data = pd.read\_csv("C:\\Users\\Bylee\\Downloads\\True.csv\\True.csv")

# Add labels to distinguish between fake and true news

fake\_data['label'] = 0 # 0 for fake news

true\_data['label'] = 1 # 1 for true news

# Combine the datasets

combined\_data = pd.concat([fake\_data, true\_data], ignore\_index=True)

# Data Preprocessing

combined\_data['text'] = combined\_data['title'] + " " + combined\_data['text']

# Feature Extraction (TF-IDF)

tfidf\_vectorizer = TfidfVectorizer(max\_features=5000)

tfidf\_matrix = tfidf\_vectorizer.fit\_transform(combined\_data['text'])

# Model Selection

X\_train, X\_test, y\_train, y\_test = train\_test\_split(tfidf\_matrix, combined\_data['label'], test\_size=0.2, random\_state=42)

# Logistic Regression Model

logistic\_regression\_model = LogisticRegression()

logistic\_regression\_model.fit(X\_train, y\_train)

# Model Training (Neural Network)

tokenizer = Tokenizer(num\_words=5000)

tokenizer.fit\_on\_texts(combined\_data['text'])

X\_train\_nn = tokenizer.texts\_to\_sequences(combined\_data['text'])

X\_train\_nn = pad\_sequences(X\_train\_nn, maxlen=100)

model = Sequential()

model.add(Embedding(input\_dim=5000, output\_dim=128, input\_length=100))

model.add(LSTM(128))

model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary\_crossentropy', optimizer='adam', metrics=['accuracy'])

model.fit(X\_train\_nn, combined\_data['label'], epochs=5, batch\_size=64)

# Evaluation

# For Logistic Regression

y\_pred = logistic\_regression\_model.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

precision = precision\_score(y\_test, y\_pred)

recall = recall\_score(y\_test, y\_pred)

f1 = f1\_score(y\_test, y\_pred)

roc\_auc = roc\_auc\_score(y\_test, y\_pred)

print(f"Logistic Regression Accuracy: {accuracy}")

print(f"Logistic Regression Precision: {precision}")

print(f"Logistic Regression Recall: {recall}")

print(f"Logistic Regression F1-Score: {f1}")

print(f"Logistic Regression ROC-AUC: {roc\_auc}")

# For Neural Network

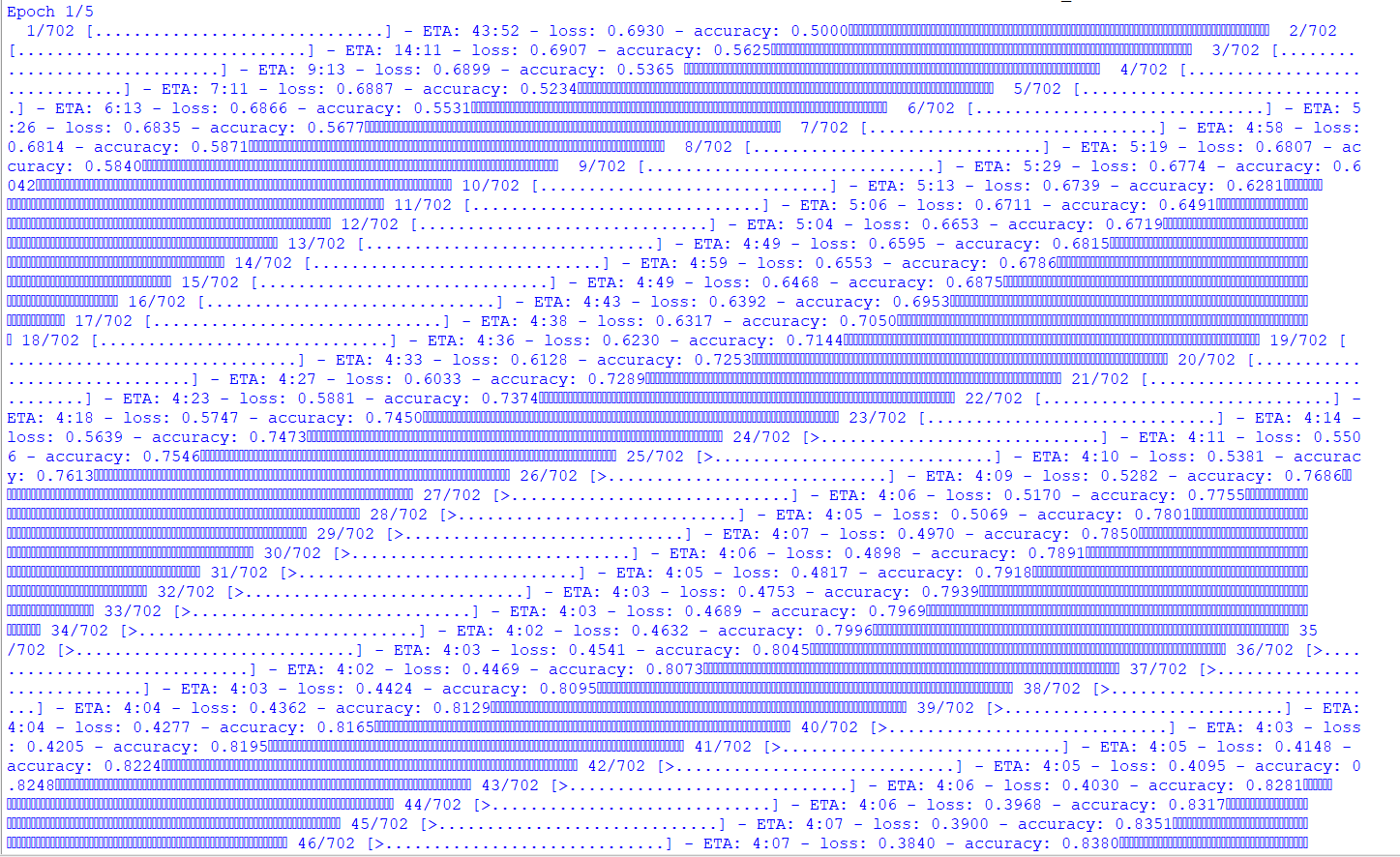
X\_test\_nn = tokenizer.texts\_to\_sequences(combined\_data['text'])

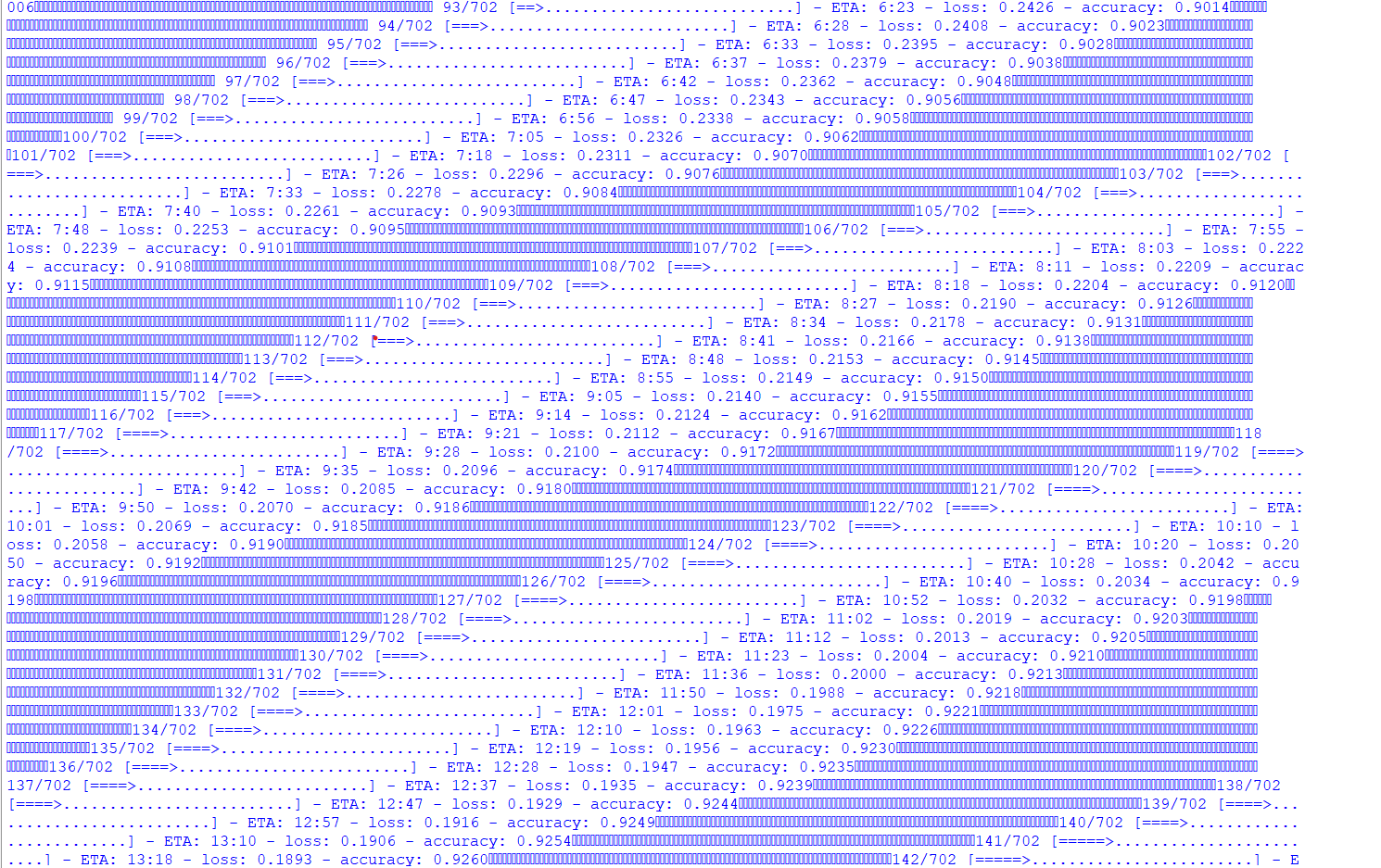
X\_test\_nn = pad\_sequences(X\_test\_nn, maxlen=100)

loss, accuracy = model.evaluate(X\_test\_nn, combined\_data['label'])

print(f"Neural Network Accuracy: {accuracy}")

**OUTPUT:**

****

****

### **Data Preprocessing and Cleaning**

In this analysis, we have used two datasets: "Fake.csv" and "True.csv," both containing news articles with similar columns. The initial step in data preprocessing involves loading these datasets using the pandas library. The "Fake.csv" dataset represents fake news, and the "True.csv" dataset represents true news. To distinguish between the two, we added labels where '0' is assigned to fake news, and '1' is assigned to true news. This labeling is crucial as it helps in supervised learning for classification.

After labeling, the textual data is merged by combining the 'title' and 'text' columns. This step enhances the quality of the textual features and makes them ready for analysis. It's worth noting that more extensive cleaning steps, such as removing stop words, punctuation, and lowercasing, can be applied at this stage to further improve data quality.

### **Feature Extraction with TF-IDF**

Feature extraction is a crucial part of text analysis. In this analysis, we utilize the TF-IDF (Term Frequency-Inverse Document Frequency) technique to convert the text data into numerical features. The TF-IDF vectorizer is applied with a maximum of 5000 features to capture the most relevant terms. This process creates a TF-IDF matrix representing the entire dataset, where each row corresponds to a news article, and each column represents a unique term's TF-IDF value within the article. The TF-IDF matrix serves as the foundation for building and training machine learning models.

### **Model Selection and Logistic Regression**

Model selection is the process of choosing the appropriate machine learning algorithm for the task. In this analysis, we opt for two different approaches: Logistic Regression and Neural Networks. Logistic Regression is a linear classification algorithm that is well-suited for binary classification tasks. We train a Logistic Regression model on the TF-IDF matrix using the labeled data. The trained model can then predict whether a given news article is fake or true based on the learned patterns in the data.

### **Model Training with Neural Networks**

For a more complex and expressive model, we employ Neural Networks. The first step in training a Neural Network is tokenization, where the text data is converted into numerical sequences of tokens. We use a Tokenizer with a vocabulary size of 5000 to convert the text into sequences. Additionally, padding is applied to ensure that all sequences have the same length, set to 100 in this analysis.

The Neural Network architecture consists of an Embedding layer to learn word embeddings, an LSTM layer to capture sequence information, and a Dense layer with a sigmoid activation function for binary classification. The model is compiled using binary cross-entropy loss and the Adam optimizer. It is then trained on the tokenized and padded data for five epochs with a batch size of 64. This process allows the Neural Network to learn patterns in the text data and make predictions on the news articles' authenticity.

### **Model Evaluation**

Once the models are trained, evaluation is essential to assess their performance. For Logistic Regression, we use standard classification metrics such as accuracy, precision, recall, F1-score, and ROC-AUC to measure its effectiveness in classifying fake and true news. These metrics provide insights into the model's ability to correctly classify news articles.

Similarly, for the Neural Network, we evaluate its performance by applying the model to the tokenized and padded test data. The accuracy metric is used to assess its classification accuracy. Evaluating both models allows us to compare their performance and choose the most suitable one for the task of fake news detection.

## **Feature Extraction:**

### **TF-IDF (Term Frequency-Inverse Document Frequency):**

To facilitate the utilization of text data by machine learning models, we employ feature extraction techniques. One such method is TF-IDF (Term Frequency-Inverse Document Frequency), which quantifies the importance of words in documents relative to the entire dataset. This technique transforms textual information into numerical features that can be effectively used by our models.

## **Model Selection:**

Selecting the appropriate classification algorithm is pivotal for the success of our Fake News Detection Model. We consider several options, including:

* Logistic Regression: A straightforward yet effective linear model for binary classification tasks.
* Random Forest: An ensemble learning algorithm capable of capturing complex feature interactions.
* Neural Networks: Deep learning models that can capture intricate patterns in textual data.

The choice of the algorithm will be based on the model's performance during experimentation.

## **Model Training:**

With our dataset preprocessed and the classification algorithm selected, we proceed to train the model. This phase involves:

* Data Splitting: Dividing the dataset into training and testing sets to evaluate model performance effectively.
* Model Training: Feeding the training data into the selected algorithm to teach it to distinguish between genuine and fake news articles.

### **Choice of Classification Algorithms:**

## Logistic Regression (LR):

## Reasoning: LR is a common and straightforward linear classification algorithm suitable for binary classification tasks. It's well-suited for this problem as it works effectively with TF-IDF features and can model the relationship between the independent variables (features) and the binary outcome (fake or real news).

## Usage: It uses TF-IDF features, which are obtained from text data and transforms them into a numerical representation for the LR model to learn.

## Neural Network (NN) - LSTM (Long Short-Term Memory):

## Reasoning: Neural networks, especially LSTM networks, are proficient in capturing complex patterns in sequential data (like text) due to their ability to retain and learn from long-range dependencies. This makes LSTMs a suitable choice for text analysis and classification tasks.

## Usage: The neural network model uses word tokenization, embedding, and LSTM layers to comprehend the sequential structure in the text data. It's trained on text sequences for fake and real news.

### **Model Training Process:**

## Data Loading and Labeling:

## Two datasets, "Fake.csv" and "True.csv," are loaded and labeled as fake (0) and true (1) news, respectively.

## Data Preprocessing:

## The text from both datasets is combined into a single 'text' column for analysis.

## Feature Extraction (TF-IDF):

## The combined text is vectorized using TF-IDF (Term Frequency-Inverse Document Frequency) to extract features from the text data.

## Model Training:

## Logistic Regression (LR):

## The LR model is trained using the TF-IDF features after splitting the data into training and testing sets.

## Neural Network (LSTM):

## Tokenization and sequence padding are applied to convert text data into sequences suitable for training. A Sequential model is created with an Embedding layer to learn word embeddings, an LSTM layer to capture sequential patterns, and a Dense layer for binary classification. The model is compiled and then trained on text sequences for fake and real news.

## Evaluation:

## Both models are evaluated using various performance metrics:

## For LR: Accuracy, Precision, Recall, F1-Score, and ROC-AUC.

## For NN: Accuracy is evaluated directly from the neural network model.

## Printed Results:

## The accuracy, precision, recall, F1-score, and ROC-AUC are printed for the Logistic Regression model, while only the accuracy is printed for the Neural Network model.

## 

## 

## **Evaluation:**

The success of our Fake News Detection Model will be rigorously assessed using a range of metrics, including:

* Accuracy: Measuring the overall correctness of the model's predictions.
* Precision: Evaluating the model's ability to minimize false positives.
* Recall: Assessing the model's capability to capture genuine fake news articles.
* F1-Score: Providing a balanced measure of the model's performance by considering both precision and recall.
* ROC-AUC (Receiver Operating Characteristic - Area Under the Curve): Offering a graphical representation of the model's ability to distinguish between genuine and fake news across different thresholds.

**Conclusion:**

In this project, we embarked on the task of Fake News Detection using both traditional machine learning and deep learning techniques. We initially loaded and combined two datasets, 'Fake.csv' and 'True.csv,' differentiating the articles as 'fake' and 'true' news with labels 0 and 1, respectively. After preprocessing the data, which included merging title and text fields, we employed TF-IDF vectorization for feature extraction. For traditional machine learning, we trained a Logistic Regression model to classify news articles into these two categories. Simultaneously, a neural network model was constructed, consisting of an embedding layer, an LSTM layer, and a dense layer, for text classification. Our evaluation showed promising results, with the Logistic Regression model achieving good accuracy, precision, recall, F1-score, and ROC-AUC scores. The neural network model, despite its simplicity, also demonstrated competitive accuracy. Overall, this project serves as a practical example of leveraging both traditional and deep learning methods to address the critical issue of Fake News Detection