**DATA SCIENCE PROJECT**

**REAL AND FAKE NEWS DETECTION**

**Introduction**

With the rise of digital media and online platforms, the spread of misinformation and fake news has become a significant concern. It is crucial to develop robust methods to distinguish between real and fake news articles to prevent the spread of false information and maintain the integrity of journalistic practices.

In this data science project, we aim to analyze a dataset consisting of real and fake news articles. The objective is to develop a machine learning model that can accurately classify news articles as either real or fake based on their content.

**Problem Statement**

This is a Natural Language Processing problem. The goal is to create a sequential classifier model that takes in news article text as input and predicts whether the news is fake or not. The tasks involved are as follows:

1. Conduct Exploratory Data Analysis on the dataset
2. Preprocess the data
3. Train a classifier that can determine the authenticity of the news
4. Evaluate the classifier using suitable evaluation metrics

**Methodology**

1. Data Preprocessing: We start by importing the dataset consisting of real and fake news articles. We perform data cleaning and preprocessing steps, including removing duplicates, punctuations, handling missing values, and filtering out non-English words and special characters. Since it was a natural language processing task, the preprocessing also included lemmatizing the text data based on English corpus.

2. Exploratory Data Analysis (EDA): We conduct exploratory data analysis to gain insights into the distribution of article lengths, common occurring words, and the distribution of news types (real vs. fake). We analyze various features from the text data, including word counts, sentence counts, character counts, word densities, and stopword counts. These features provide additional information that can improve the performance of our machine learning models. Visualization techniques such as histograms and word clouds are used to visualize the data.

3. Feature Extraction: We extract the main text and its corresponding label (1 for true and 0 for fake) to train the model.

4. Model Building: We build two machine learning models for binary classification: logistic regression and bi-directional LSTM, a type of recurrent neural network (RNN). Logistic regression is a traditional machine learning algorithm, while RNN is a deep learning model capable of capturing sequential information in text data.

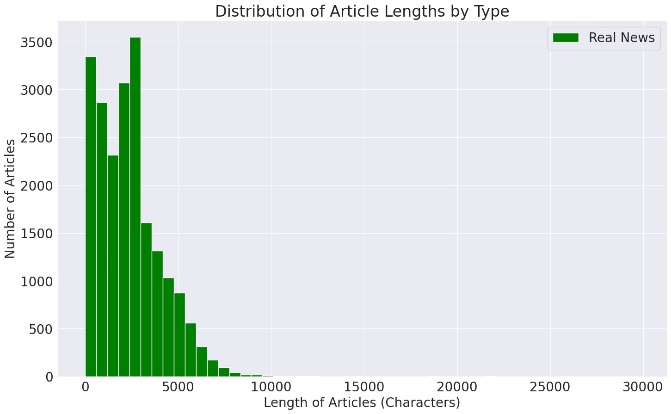
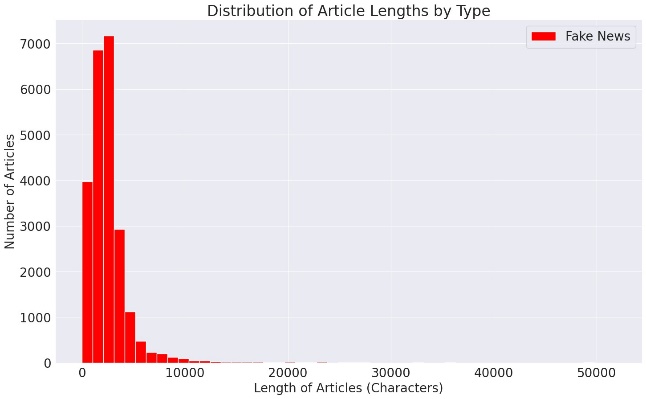
The model architecture is a sequential neural network consisting of several layers:

* An embedding layer that converts input sequences into dense vectors of dimensionality 128.
* Two bidirectional LSTM layers, each with 128 units, allowing the model to capture both forward and backward context information.
* A dense layer with 64 units and ReLU activation function.
* A dropout layer to prevent overfitting by randomly dropping 50% of the units.
* A final dense layer with a single unit and sigmoid activation function for binary classification.

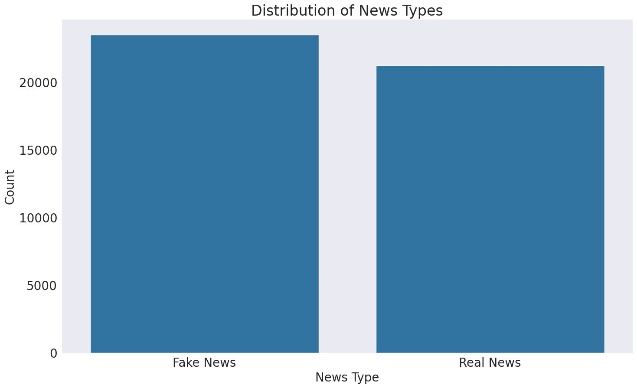
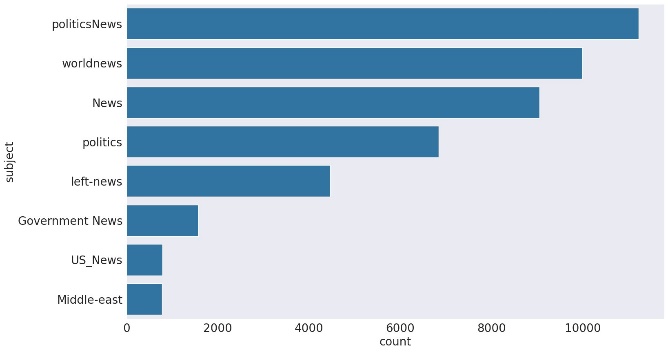
The model training process is configured with early stopping, a technique that halts training when the validation loss fails to improve for a certain number of epochs (patience set to 2 in our case). This prevents overfitting and saves computational resources by restoring the best weights observed during training. The model is compiled using binary cross-entropy loss, suitable for binary classification tasks, and optimized using the Adam optimizer with a learning rate of 1e-4. During training, the model is fit to the training data for 10 epochs, with a validation split of 0.1 and a batch size of 30. Additionally, the training process is monitored by the early stopping callback to ensure optimal performance while avoiding overfitting.

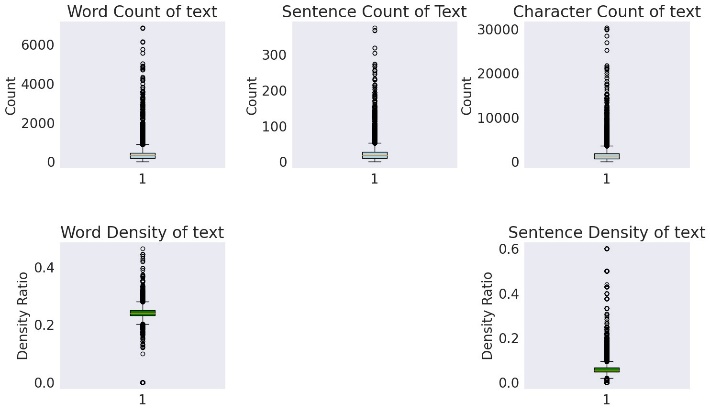
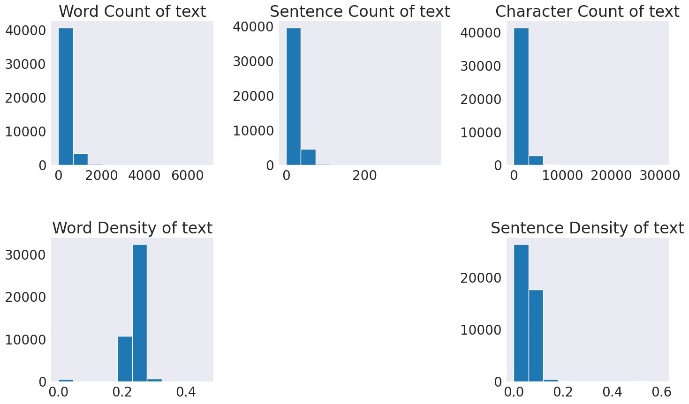
5. Model Evaluation: We evaluate the performance of the models using metrics such as accuracy, precision and recall. Additionally, we visualize the confusion matrix to analyze the model's performance in predicting real and fake news articles.

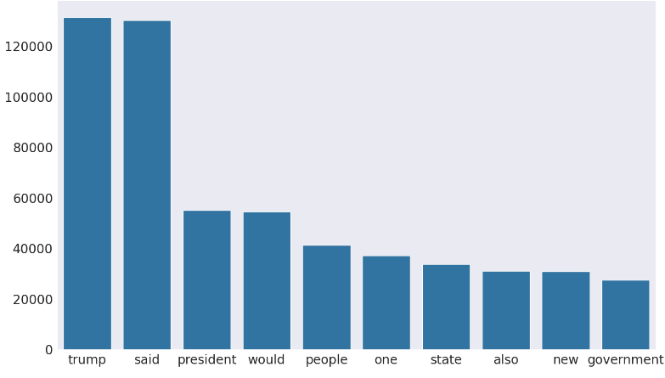
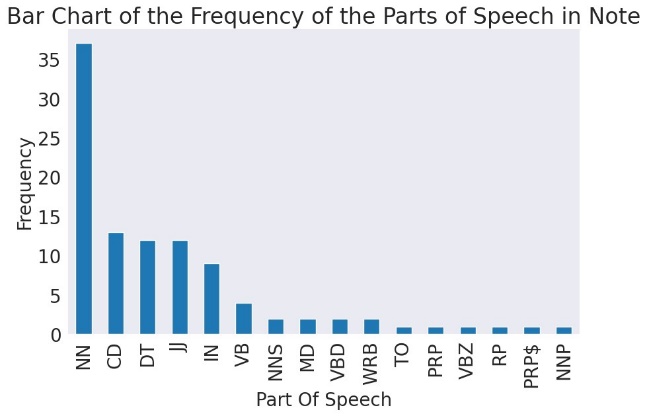
**Visualization Outputs**

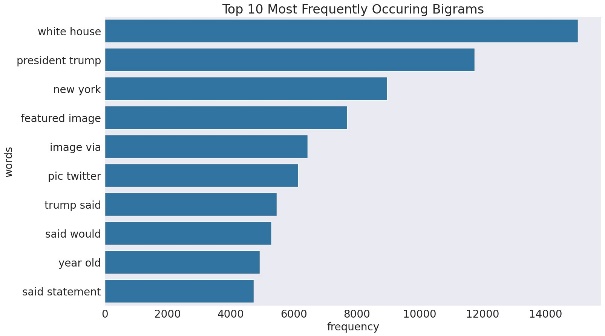
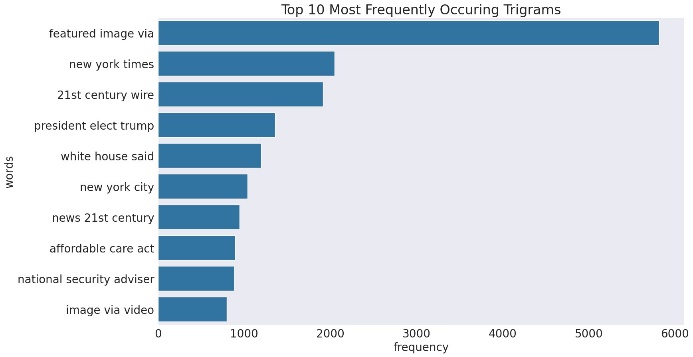


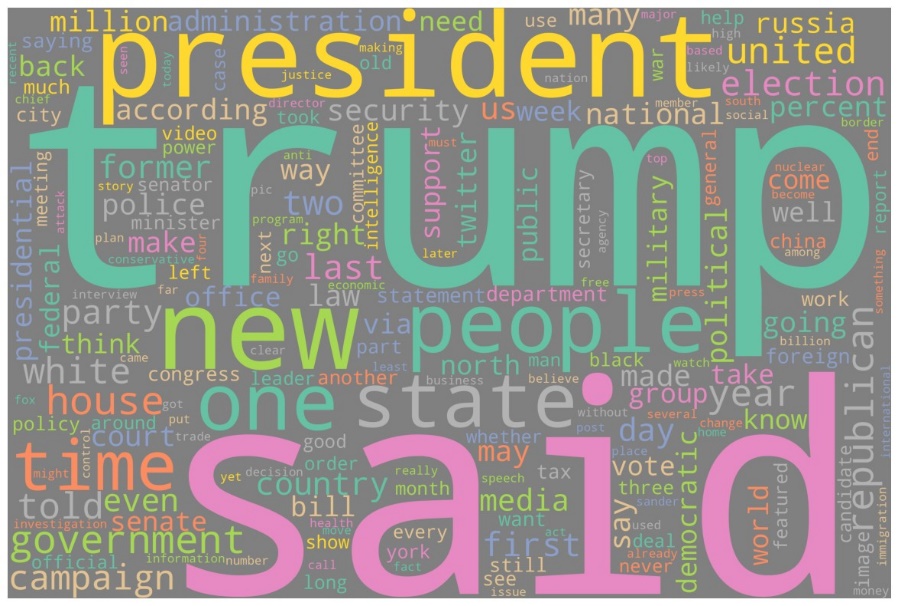
- =









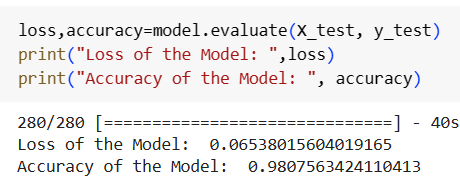


**Results**

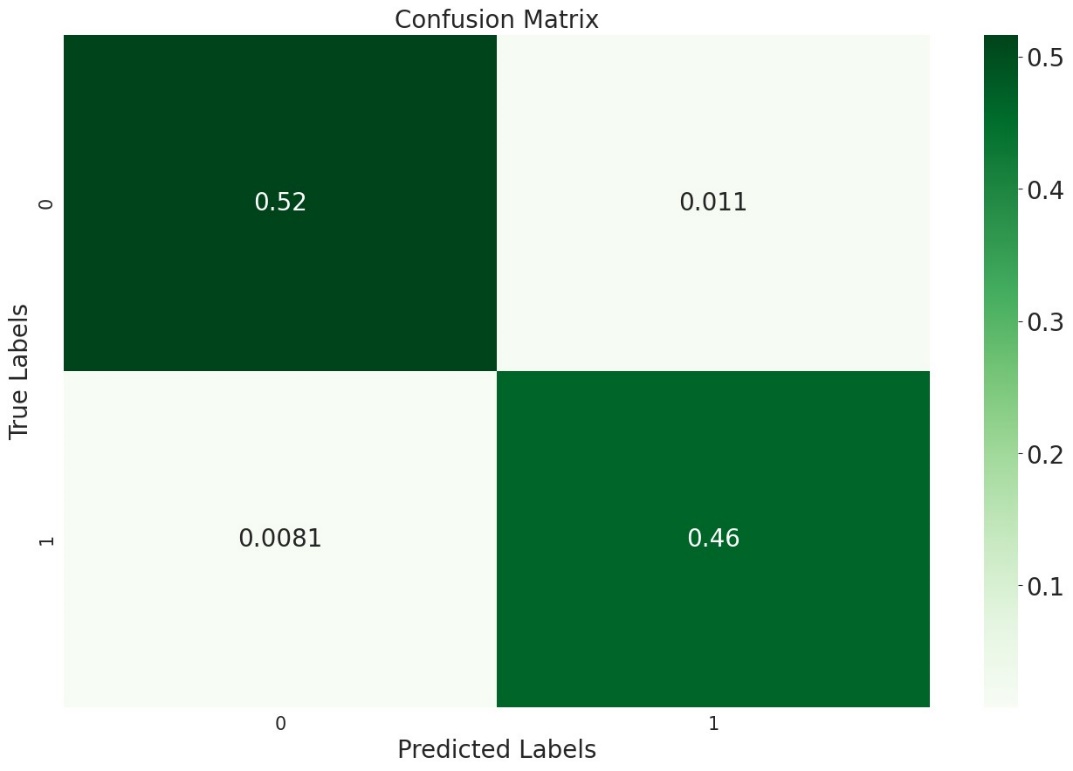
- Logistic Regression Model: The logistic regression model achieved a training accuracy of 0.9801124444071494 and a testing accuracy of 0.9674423808458268.

- LSTM Model: The LSTM model achieved a testing accuracy of 0.9807563213246812 with a precision of 0.9764650506001412 and a recall of 0.9829424307036247 on the test dataset.

Overall the model showed a loss of 6.5 percent and the accuracy of the model is 98.07 percent.



- Confusion Matrix: The confusion matrix provides insights into the model's performance in predicting real and fake news articles. The matrix shows the distribution of true positive, true negative, false positive, and false negative predictions. 52 percent of the predicted labels are True Negatives which means the model predicted fake news accurately. Likewise, 46 percent of the labels are True Positives. This suggests that 46 percent of the times the model predicted positive for real news and it’s true.



Overall, the results demonstrate the effectiveness of the machine learning models in accurately classifying news articles as real or fake based on their content. We used logistic regression model in comparison with LSTM Model and although both models showed great accuracy, LSTM is a deep learning model whose ability to retain long-term dependencies and handle input sequences of varying lengths makes them well-suited for tasks like natural language processing. In summary, LSTM networks offer efficiency and benefits in modeling sequential data, making them a powerful tool for predicting fake news.