Semantic Classification Assignment

Fake News Detection

Submitted by – Poonam Parate and Pankaj Kumar Agrawal

Problem Statement:

The spread of fake news has become a significant challenge in today's digital world. With the constant flow of news articles published daily, it is becoming harder to distinguish between credible and misleading information. This creates a need for systems that can automatically classify articles as true or fake, helping reduce misinformation and protect public trust.

Objective:

The objective of this assignment is to develop a Semantic Classification model. You will be using Word2Vec method to extract the semantic relations from the text and develop a basic understanding of how to train supervised models to categorise text based on its meaning, rather than just syntax. You will explore how this technique is used in situations where understanding textual meaning plays a critical role in making accurate and efficient decisions.

In this assignment, you will develop a Semantic Classification model that uses the Word2Vec method to detect recurring patterns and themes in news articles. Using supervised learning models, the goal is to build a system that classifies news articles as either fake or true.

Tasks:

1. **Data Preparation:**
    1. Add new columns
    2. Merge Data Frames
    3. Handle null values
    4. Merge the relevant columns

In the data preparation phase, we first added a news_label column to both True and Fake news datasets to indicate whether a news item is real (1) or fake (0). We then merged these two Data Frames into one combined dataset. To ensure data quality, we handled missing values by dropping rows with nulls in important columns like title, text, or date. Finally, we merged relevant columns (like title and text) into a single news text column for consistent text processing.

2. **Text Preprocessing:**

1. Text cleaning
2. POS tagging and lemmatisation

In the text preprocessing step, we cleaned the news content by converting it to lowercase, removing punctuation, special characters, numbers, and extra whitespace. We then performed Part-of-Speech (POS) tagging and lemmatization using SpaCy, focusing only on meaningful words (e.g., nouns) while filtering out stopwords and irrelevant POS tags like pronouns and conjunctions. This helped reduce noise and preserve the semantic structure of the text. The cleaned and lemmatized text was stored in new columns for further analysis.

3. **Train Validation Spilt**

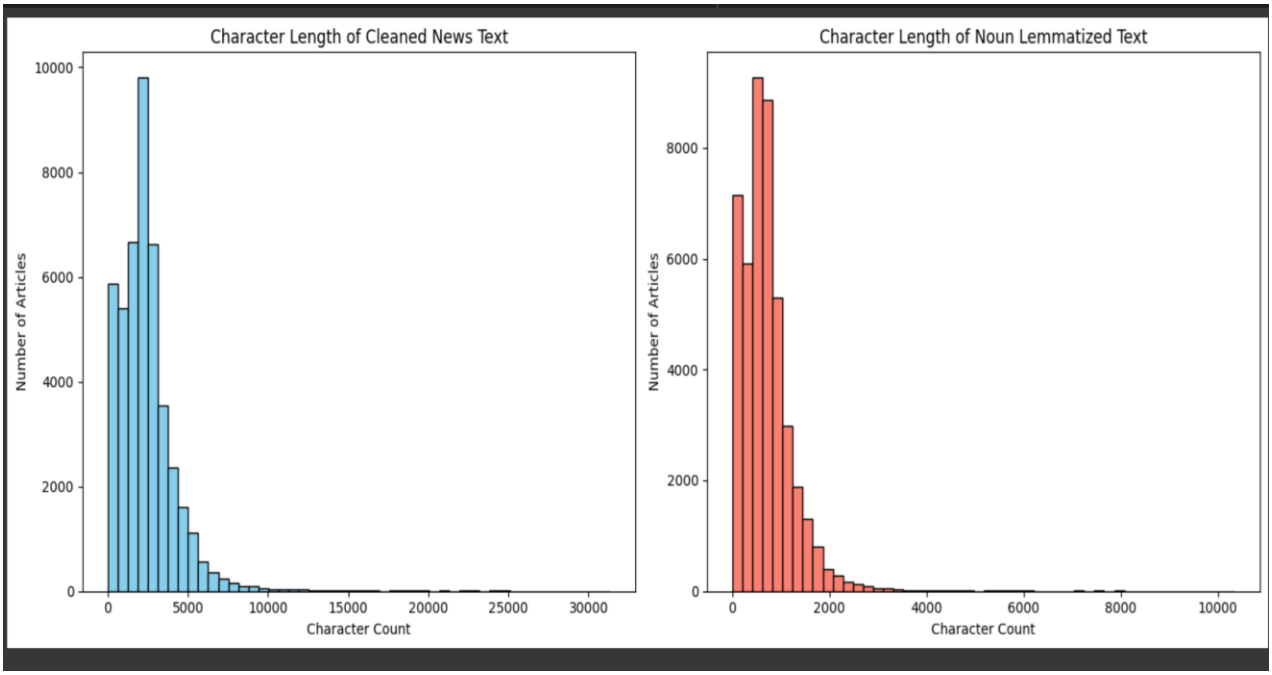
1. Split the data into training and validation sets with a 70:30 ratio

We split the processed dataset into training and validation sets using a 70:30 ratio to ensure the model learns from a majority of the data while being evaluated on unseen data. This helps in assessing the model's generalization performance and avoiding overfitting.

4. Exploratory Data Analysis on Training Data and Validation Data

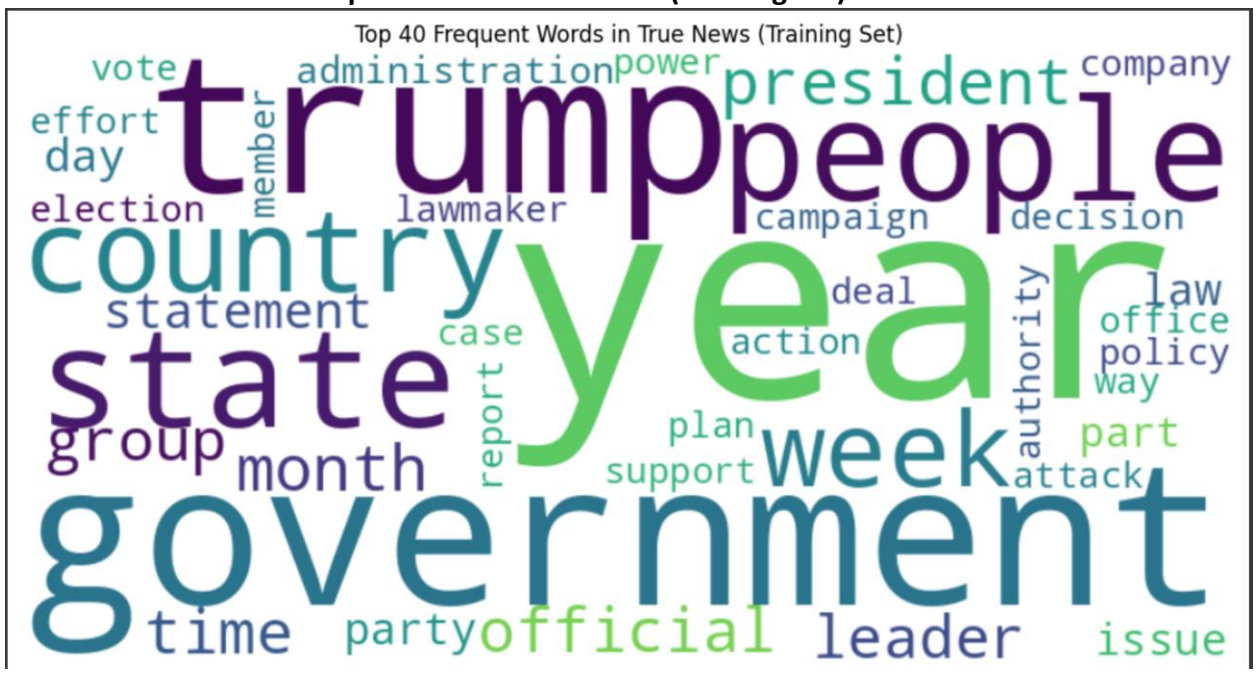
1. Visualise character lengths of cleaned news text and lemmatised news text with POS tags removed

Plotted histogram plot to visualise character lengths



2. Find and display the top 40 words by frequency among true and fake news

Top 40 words in True New (Training Set)



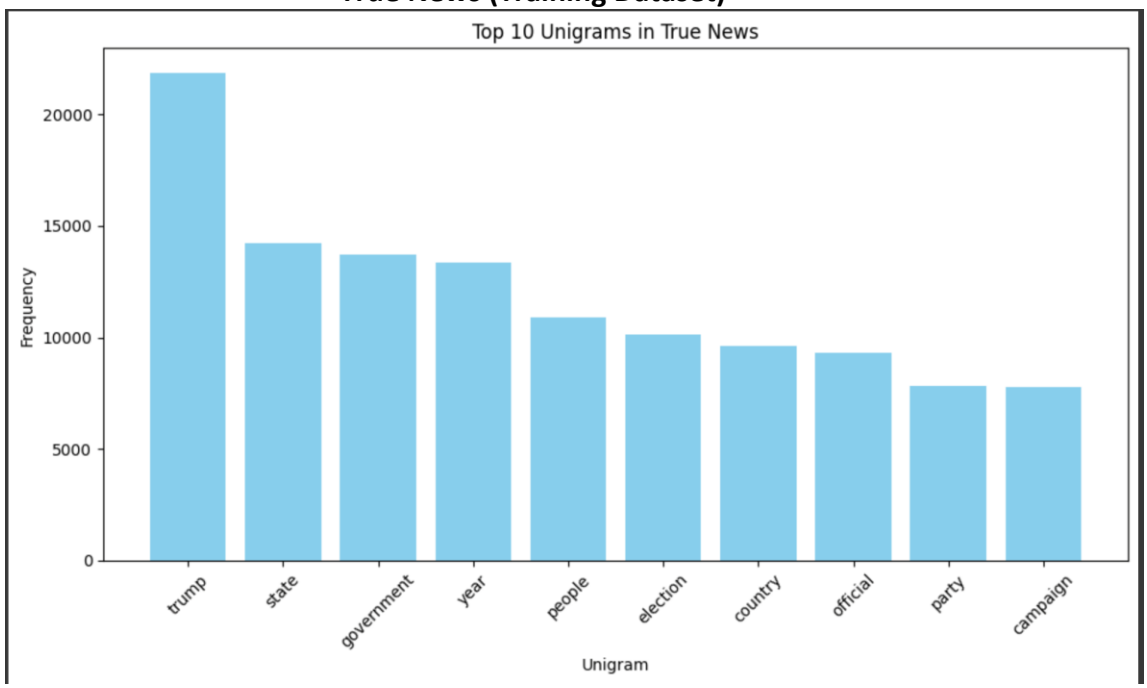
Most Prominent Words: trump, government, year, state, country, people, official, leader, week, policy

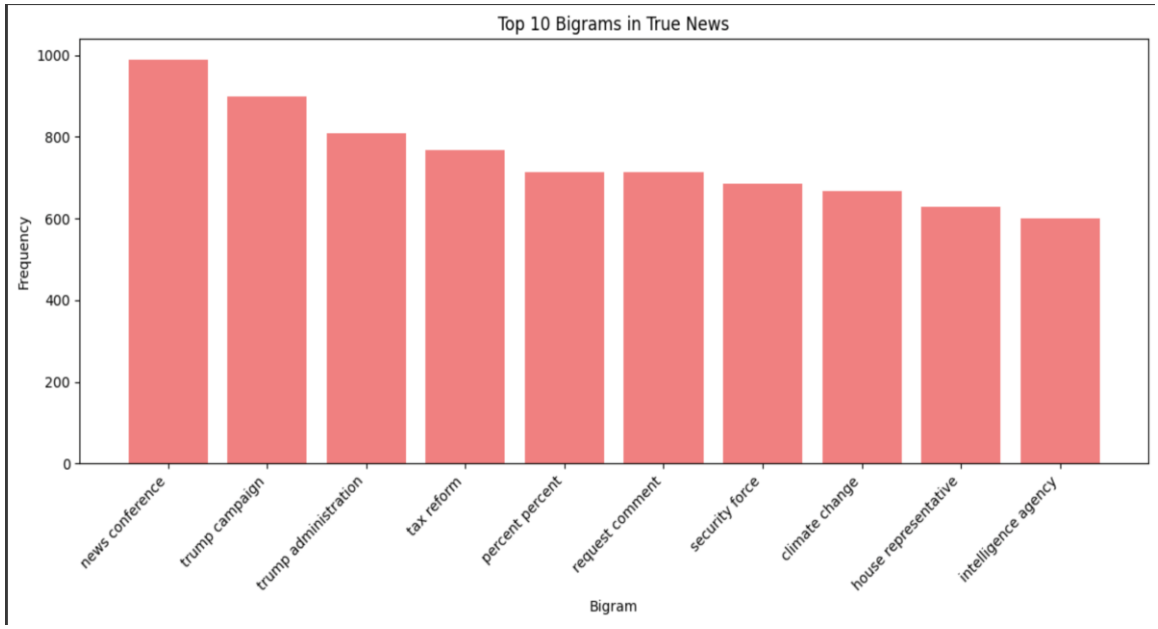
Top 40 words in FakeNew (Training Set)

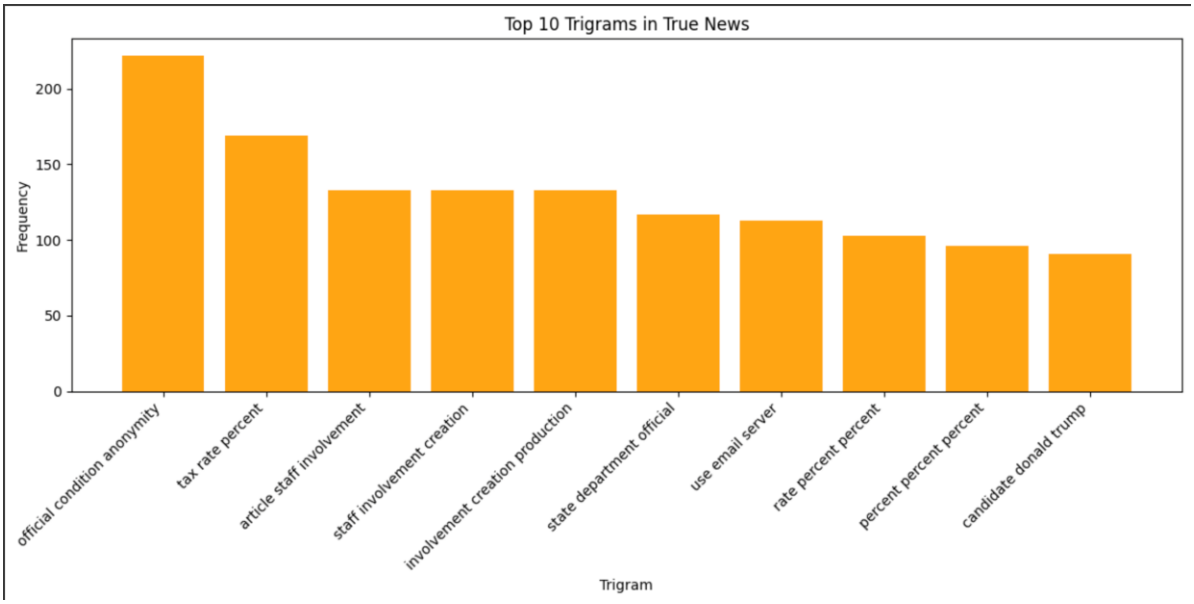
Most Prominent Words: trump, people, time, video, thing, fact, woman, story, medium, family

3. Find and display the top unigrams, bigrams and trigrams by frequency in true and fake news

True News (Training Dataset)

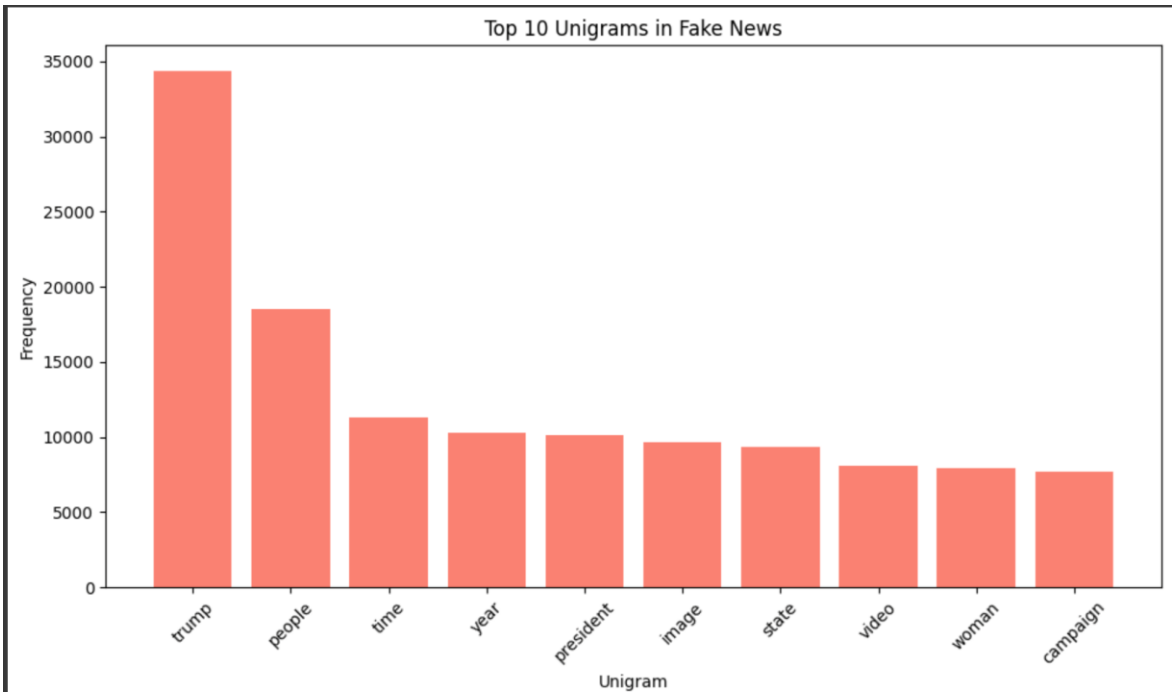


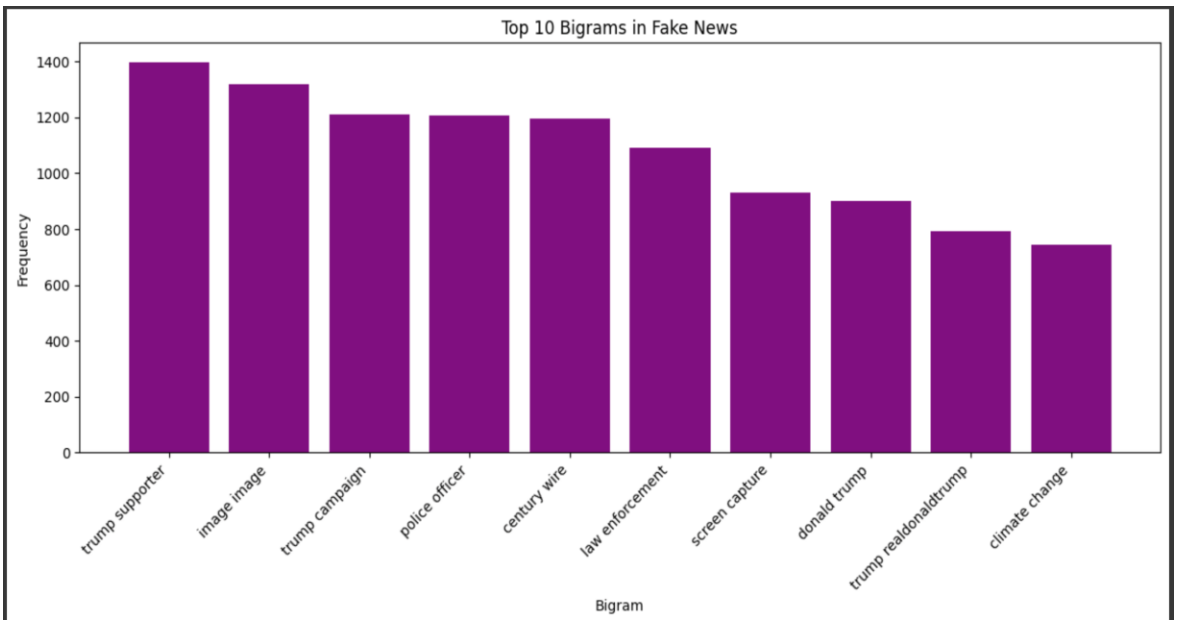


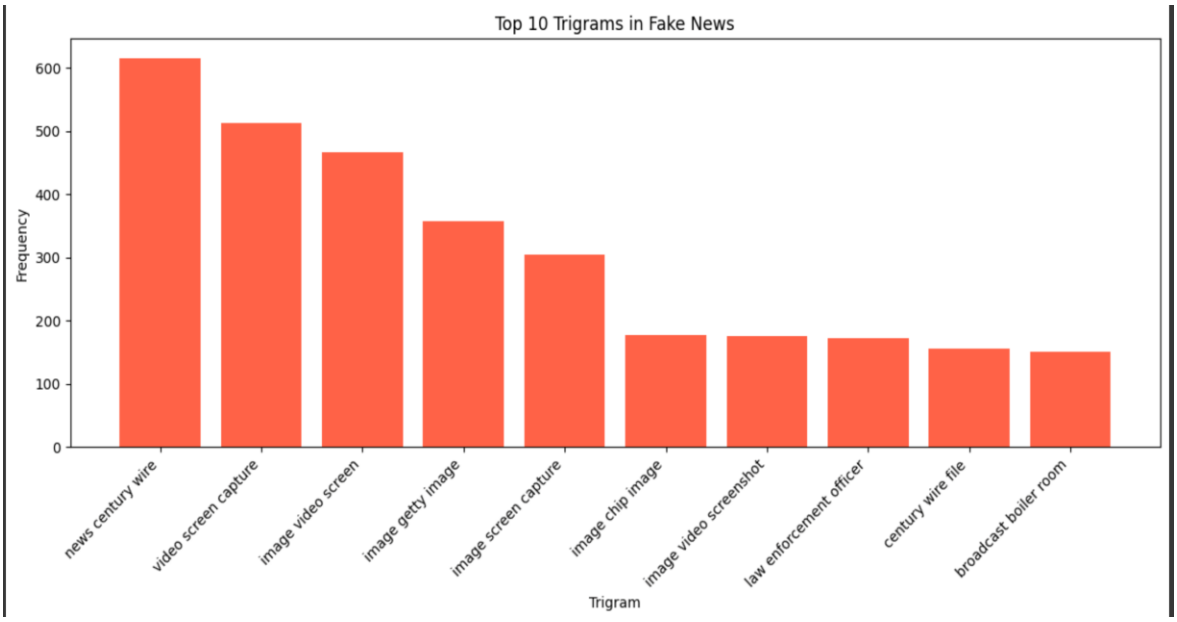


- **"trump"** appears **over 21,000 times**, making it the most dominant word in the true news set.
- Other high-frequency words like **"state," "government," "year," "people," "election,"** and **"country"** indicate that true news articles often focus on **politics, governance, and national affairs**.

Fake News (Training Dataset)







- **"trump"** dominates significantly with **over 34,000 occurrences**, indicating that fake news content heavily revolves around topics related to Trump - possibly political conspiracies or misinformation.
- Words like **"people"**, **"time"**, and **"year"** also appear frequently, showing a general or vague temporal and societal context often used to make fake stories sound urgent or relatable.

In the EDA phase, we first analysed the character lengths of both the cleaned and lemmatized news text to understand text size distribution across samples. We then visualized the most frequent 40 words in fake news using a word cloud, helping identify common patterns and themes. Additionally, we extracted and plotted the top unigrams, bigrams, and trigrams for both true and fake news articles to uncover significant word combinations and phrase patterns. These insights helped differentiate linguistic characteristics between real and fake news, guiding feature selection for model building.

5. Feature Extraction

1. Initialise the Word2Vec model
2. Extract vectors for cleaned news data

We initialized the pre-trained Word2Vec model (word2vec-google-news-300) to capture semantic relationships between words. For each news article, we computed the average Word2Vec vector by aggregating word embeddings from the cleaned and lemmatized text, converting unstructured text into numerical features for model training.

6. Model Training and Evaluation

We trained three supervised classification models - Logistic Regression, Decision Tree, and Random Forest - using the Word2Vec vectorized text data. Each model was trained on the training set and evaluated on the validation set to assess its predictive performance. We used metrics such as accuracy, precision, recall, and F1-score to compare model effectiveness. The evaluation helped identify which model generalized best for classifying fake and true news based on semantic features.

1. Build the logistic regression model

Classification Report of logistic regression:

	precision	recall	f1-score	support
Fake News	0.91	0.90	0.90	7039
True News	0.89	0.90	0.89	6425
accuracy			0.90	13464
macro avg	0.90	0.90	0.90	13464
weighted avg	0.90	0.90	0.90	13464

The model achieved **90% accuracy**, showing balanced performance in detecting both fake and true news with high precision and recall.

2. Build the decision tree model

```
Classification Report for Decision Tree Model:

              precision    recall  f1-score   support

   Fake News       0.82      0.86      0.84      7039
   True News       0.83      0.79      0.81      6425

    accuracy                           0.82     13464
   macro avg       0.82      0.82      0.82     13464
weighted avg       0.82      0.82      0.82     13464
```

The Decision Tree model achieved **82% accuracy**, with slightly higher recall for fake news and better precision for true news.

3. Build the random forest model

```
Classification Report for Random Forest Model:

              precision    recall  f1-score   support

   Fake News       0.90      0.92      0.91      7039
   True News       0.91      0.89      0.90      6425

    accuracy                           0.91     13464
   macro avg       0.91      0.91      0.91     13464
weighted avg       0.91      0.91      0.91     13464
```

The Random Forest model achieved **91% accuracy**, showing strong and balanced performance across both fake and true news classifications.

Conclusion:

1. True news used more formal and topic-specific words (like government or policy), while fake news often used emotional or dramatic words (like video, image, law) to catch attention.
2. We used Word2Vec to understand the meaning of words in context, which helped the model figure out the difference between real and fake news more accurately.
3. Out of the three models we tested, the **Random Forest model** worked the best, giving us 91% accuracy.
4. We focused on accuracy and F1-score, which helped us check how well the model avoided wrong predictions for both fake and real news.
5. The graphs and word clouds showed that fake and real news use very different kinds of language, which helped us design better features for our model.
6. Using word meaning and a strong model like Random Forest made it easier to spot fake news, which is helpful for fighting misinformation online.