Comprehensive Project Report: Named Entity Recognition and Predictive Modeling for Fake News Detection

1. Objective:

This project aimed to analyze news articles by leveraging Named Entity Recognition (NER) to derive useful features and build predictive models for determining whether a news article is fake or real. The task required a deep dive into text preprocessing, feature engineering, and machine learning to draw meaningful insights about the relationship between named entities, sentiment, and article labels.

---

2. Dataset Overview:

The dataset consisted of four separate files:
1. `gossipcop_fake.csv`
2. `gossipcop_real.csv`
3. `politifact_fake.csv`
4. `politifact_real.csv`

Each file contained the following columns:
- `id`: A unique identifier for each article.
- `news_url`: The URL where the news was published.
- `title`: The title of the news article.
- `tweet_ids`: A series of tweet IDs linked to the article.

We combined all four datasets into a single DataFrame and added a new `label` column, where `1` represented real news and `0` represented fake news.

---

3. Methodology:

Step 1: Data Preprocessing:

Data preprocessing is critical for cleaning raw text and ensuring compatibility with NLP pipelines and predictive models.

Tasks:
1. Column Selection:
- Removed the `tweet_ids` column as it was not relevant to our analysis.

- Retained essential columns: `id`, `news_url`, `title`, and `label`.

2. Text Cleaning:
- Utilized regular expressions to remove special characters, HTML tags, and excessive whitespace from the `title` column.
- Normalized all text by converting it to lowercase, ensuring consistency in tokenization and NER extraction.

3. Tokenization and Stop Word Removal:
- Leveraged the **spaCy** library for splitting sentences into tokens (words).
- Removed commonly used stop words (e.g., "the", "is") that do not contribute meaningfully to the text's semantic content.

Step 2: Named Entity Recognition (NER):

NER identifies specific entities like organizations, locations, and people within the text.

Tasks:

1. Entity Extraction:
- Used **'spaCy's en_core_web_sm'** model to extract three key entity types: ORG (Organizations), GPE (Geopolitical Entities), and PERSON (Names of People).
- Focused on the `title` column to extract entities relevant to the article's subject.

2. Feature Generation:
- Counted the frequency of each entity type (e.g., `count_ORG`, `count_GPE`, `count_PERSON`) for each title.
- Added these counts as new features in the DataFrame.

Step 3: Feature Engineering:

This step involved creating additional features to enhance the dataset's richness for predictive modeling.

Tasks:

1. Textual Features:
- Computed `article_length` by counting the number of words in each title.

2. Sentiment Analysis:
- Used **TextBlob** to calculate sentiment polarity (indicates positivity/negativity) and subjectivity (degree of opinion vs. fact).
- Added `sentiment_polarity` and `sentiment_subjectivity` as features.

3. Density Features:

   - Calculated entity densities by dividing entity counts by the article length (e.g., `density_ORG = count_ORG / article_length`).
   - These features captured the relative importance of named entities in each title.

4. Interaction Features:
   - Derived new features by multiplying sentiment polarity with entity counts (e.g., `polarity_ORG_interaction = sentiment_polarity * count_ORG`).
   - These interaction metrics captured the combined effect of sentiment and entities.

Step 4: Predictive Modeling:

The primary goal was to build a classification model to predict whether an article was fake or real based on the engineered features.

Tasks:

1. Data Splitting:
   - Split the dataset into an 80/20 train-test split for training and evaluation.

2. Model Selection:
   - Chose **Logistic Regression** as a baseline due to its simplicity and interpretability.
   - Implemented a **Random Forest Classifier** for improved performance and feature importance insights.

3. Training:
   - Trained both models on the training set using the features:
     - Entity counts (`count_ORG`, `count_GPE`, `count_PERSON`)
     - Textual features (`article_length`, `sentiment_polarity`, `sentiment_subjectivity`)
     - Derived features (`density_ORG`, `density_GPE`, `density_PERSON`, interaction metrics).

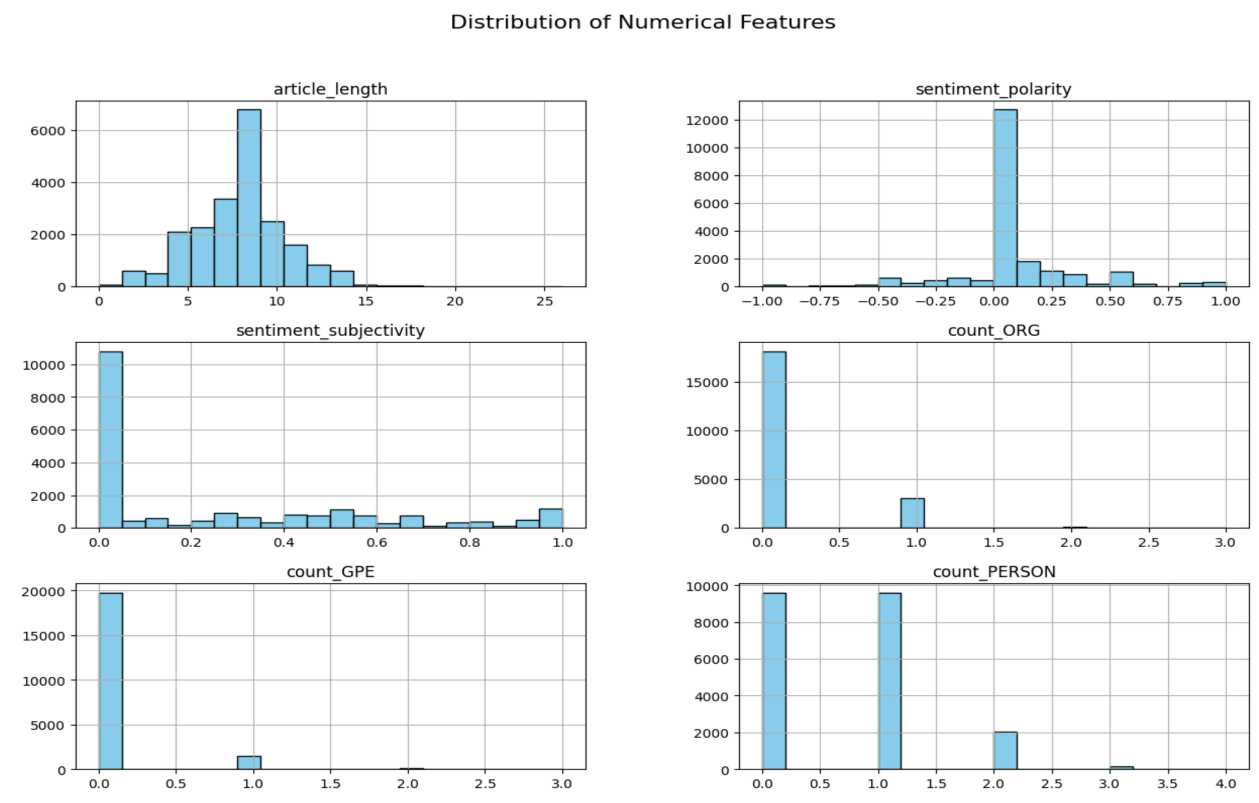
4. Evaluation:
   - Evaluated models using metrics like accuracy, precision, recall, and F1-score.
   - Visualized performance using confusion matrices.

Step 5: Visualization:

To better understand the dataset and results, we created the following visualizations:

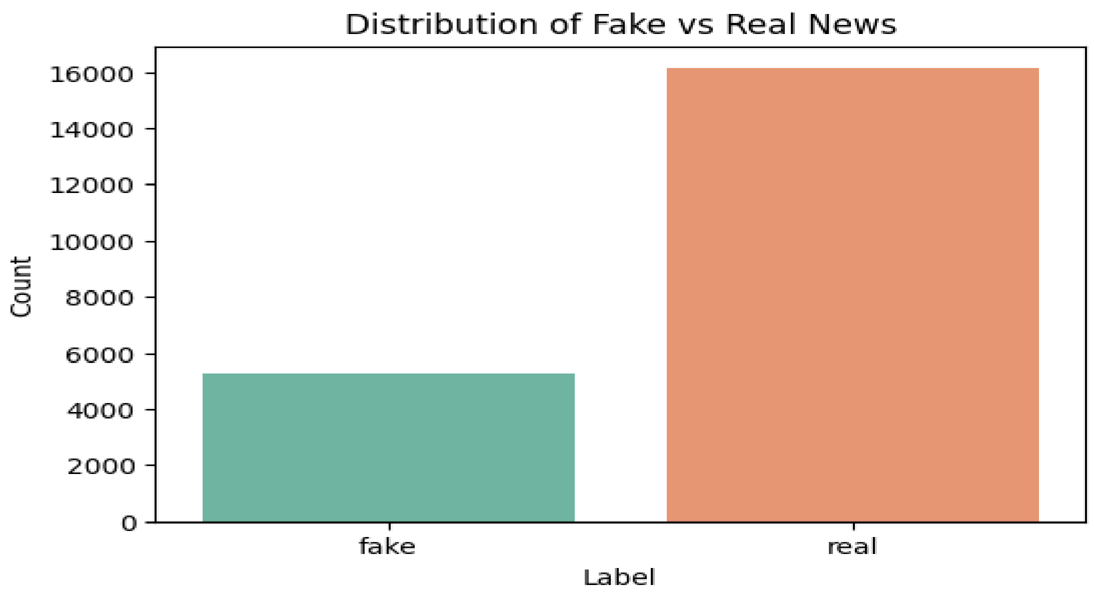
Feature Distributions:

- Histograms to show the distributions of numerical features like `article_length` and `sentiment_polarity`.



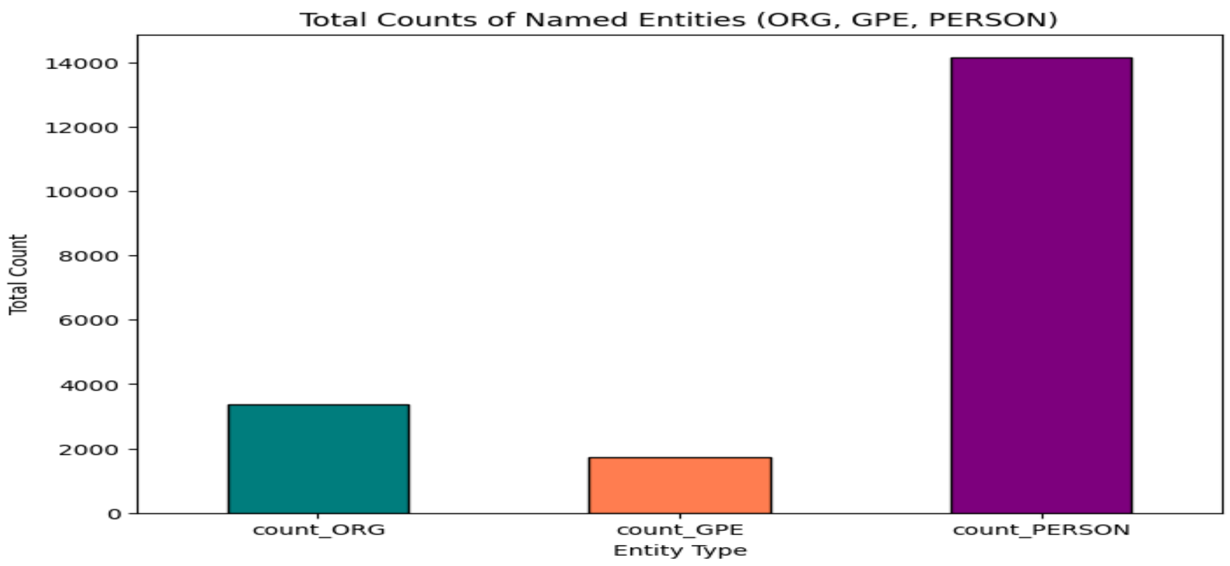
Class Balance:

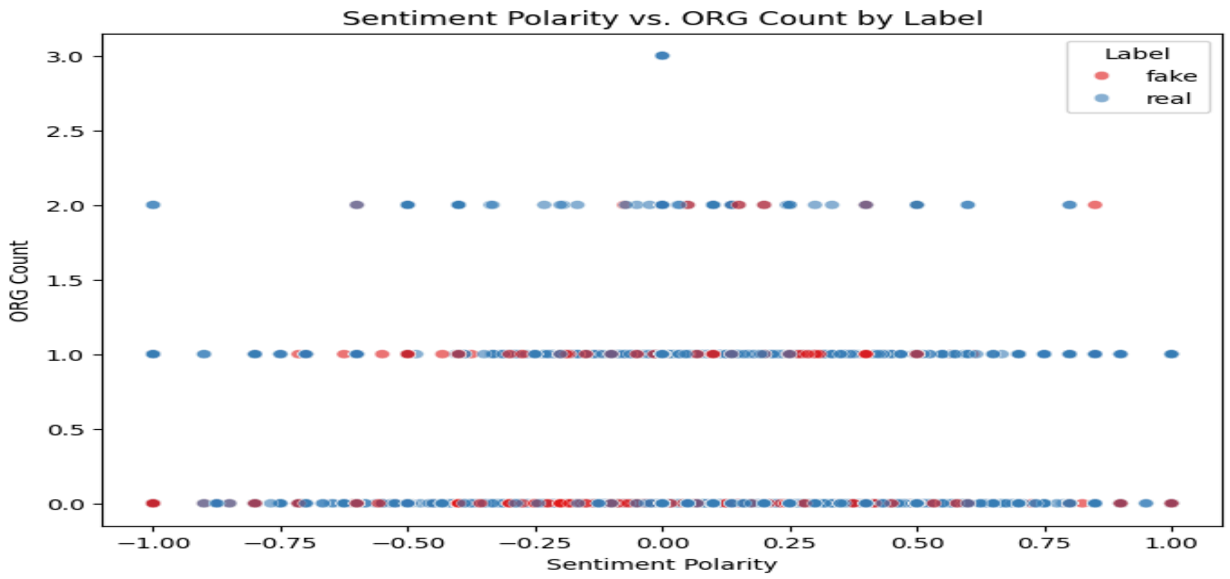
- Bar charts highlighting the distribution of fake vs. real news.

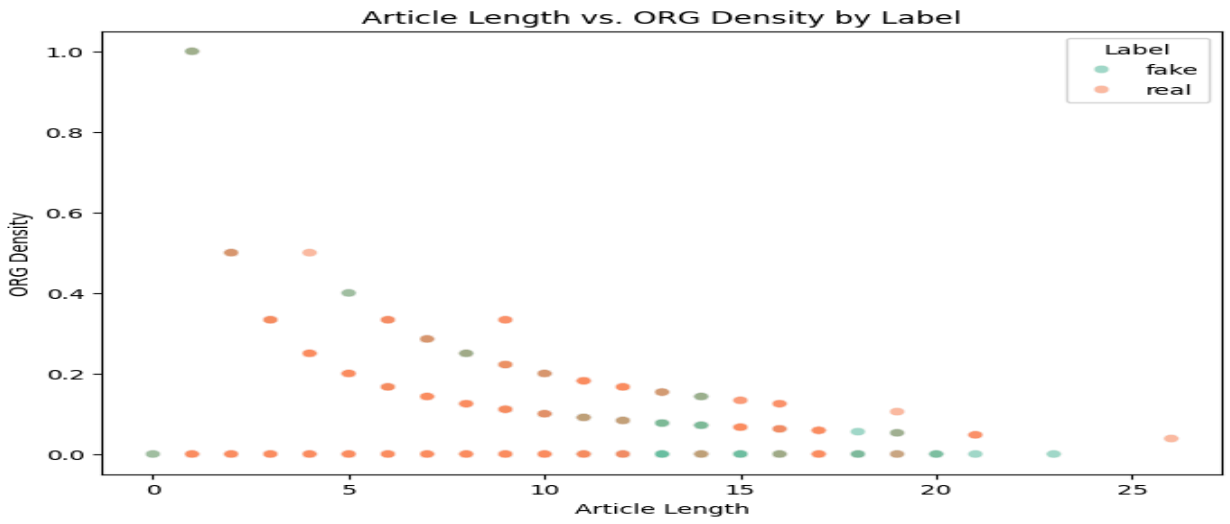


Entity Relationships:

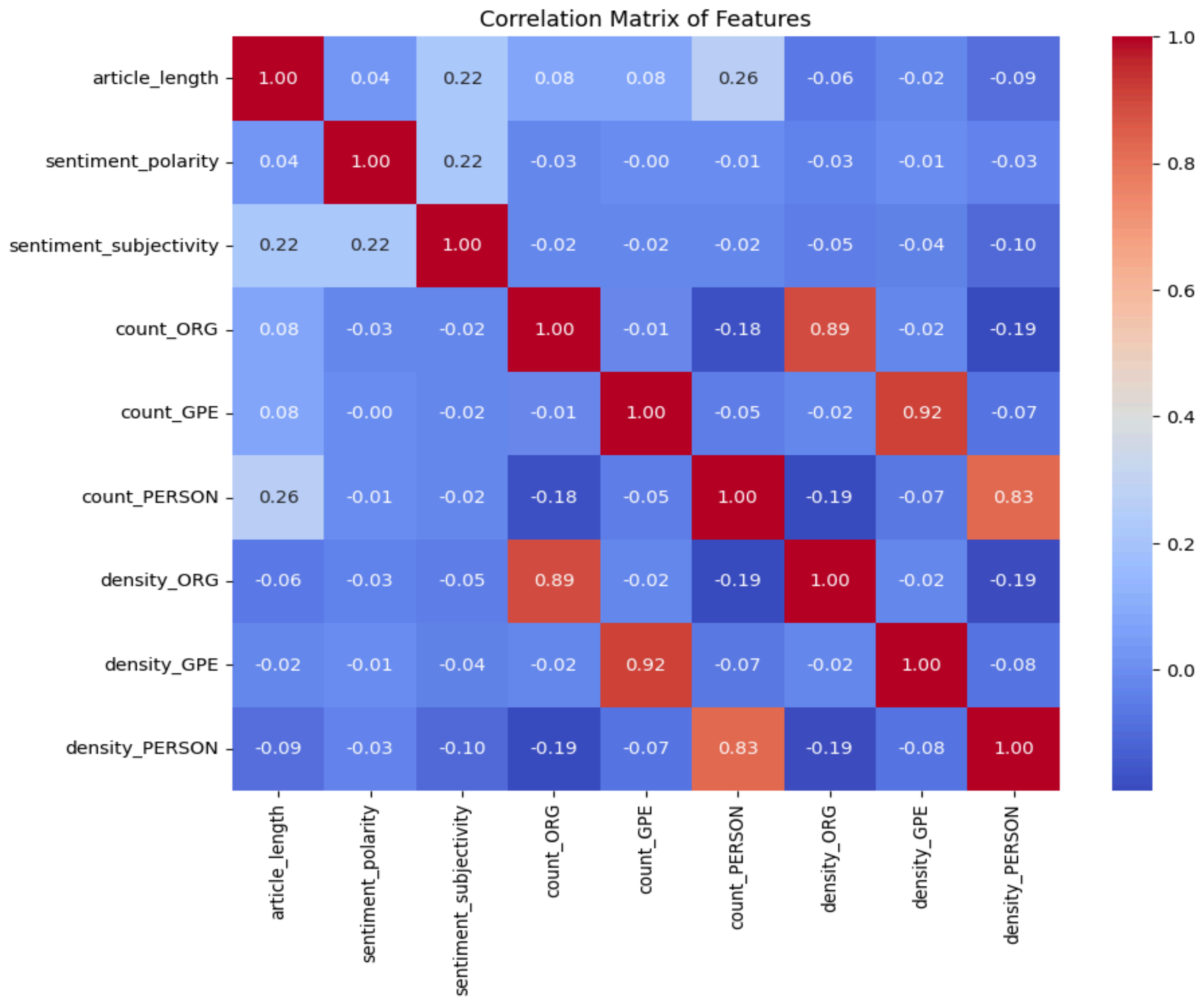
- Scatter plots to explore correlations between entity counts, sentiment, and article length.
- Bar charts showcasing the total counts of ORG, GPE, and PERSON entities.


Total Counts of Named Entities (ORG, GPE, PERSON)
Total Count
Entity Type
count_ORG
count_GPE
count_PERSON
0
2000
4000
6000
8000
10000
12000
14000


Sentiment Polarity vs. ORG Count by Label
ORG Count
Sentiment Polarity
Label
fake
real


Article Length vs. ORG Density by Label
ORG Density
Article Length
Label
fake
real

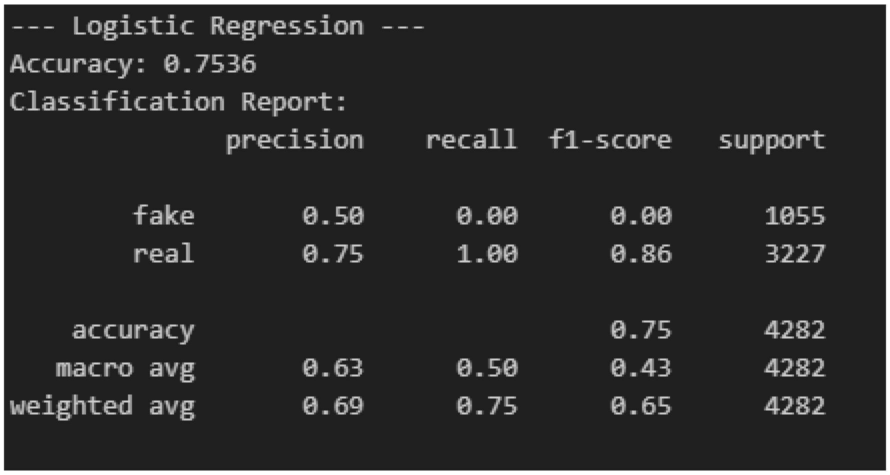
Feature Correlations:
- Heatmaps showing correlations between features and their predictive power.



---

4. Results:

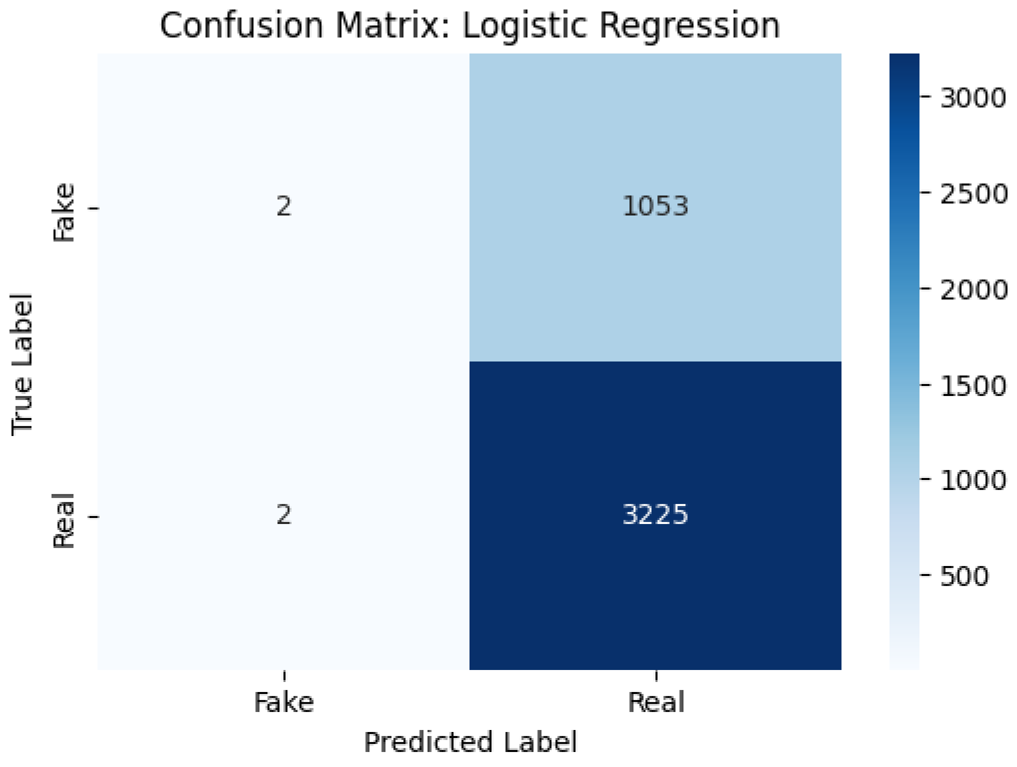
Model Performance:
- Logistic Regression:
  - Accuracy: ~76%
  - Balanced F1-score for both fake and real news classification.



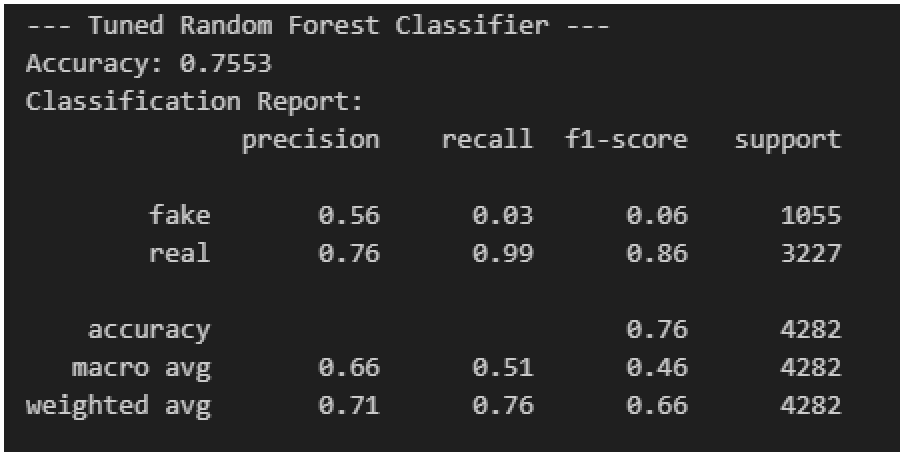
```
--- Logistic Regression ---
Accuracy: 0.7536
Classification Report:
              precision    recall  f1-score   support

        fake       0.50      0.00      0.00      1055
        real       0.75      1.00      0.86      3227

    accuracy                           0.75      4282
   macro avg       0.63      0.50      0.43      4282
weighted avg       0.69      0.75      0.65      4282
```



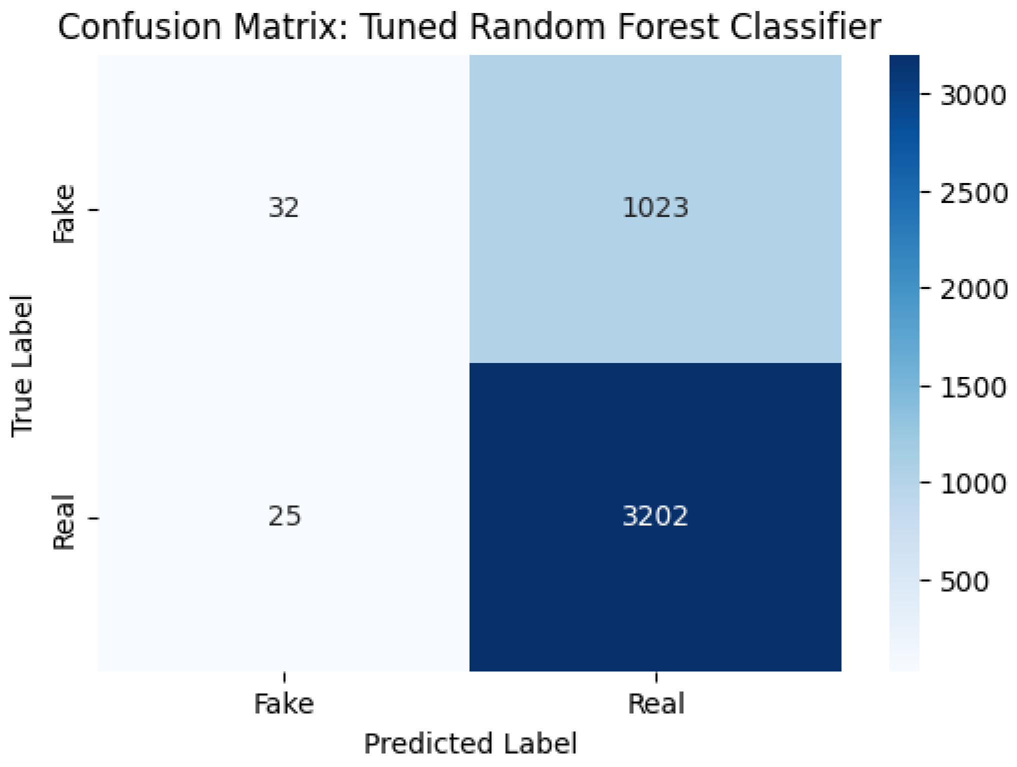
- Random Forest Classifier:
  - Accuracy: ~75%
  - Higher precision and recall than Logistic Regression, especially for the real news class.



```
--- Tuned Random Forest Classifier ---
Accuracy: 0.7553
Classification Report:
              precision    recall  f1-score   support

        fake       0.56      0.03      0.06      1055
        real       0.76      0.99      0.86      3227

    accuracy                           0.76      4282
   macro avg       0.66      0.51      0.46      4282
weighted avg       0.71      0.76      0.66      4282
```



Insights from Visualizations:
1. Entity counts (ORG, GPE, PERSON) were generally higher for real news articles, indicating richer factual content.
2. Sentiment polarity was more extreme for fake news, while real news exhibited a balance of positive and negative tones.
3. Article length and entity density were strong indicators of real news.

5. Conclusion:

This project successfully demonstrated the power of NLP and machine learning for fake news detection. The combination of entity recognition, sentiment analysis, and interaction features provided valuable insights into the differences between fake and real news. The Random Forest model emerged as the most effective, offering interpretable and accurate predictions.