NLP Project - Fake News

Mauricio & Kepa

1. Data Loading and Exploration

- A dataset of approximately 40,000 news articles (50% real, 50% fake).
- Main columns: **label**, **title**, **text**, subject, and date.
- Initial exploration: checked data structure, missing values, and label distribution.

2. Preprocessing

- Text cleaning: converted to lowercase, removed punctuation, numbers, and stopwords.
- Lemmatization.
- TF-IDF vectorization (with bigrams), with vocabulary of about 8,500 relevant words.

3. Feature Engineering

- TF-IDF features with an n-gram range of (1,2) to capture word combinations.
- Filtered out biased and overly rare words to improve model generalization and avoid data leakage.

.

4. Models

Model	Accuracy	Observation
Logistic Regression	0.877	Good balance between precision and recall
Naive Bayes	0.839	Slight drop in recall performance
Random Forest	**0.881**	**Best overall performance and stability**

5. Key results

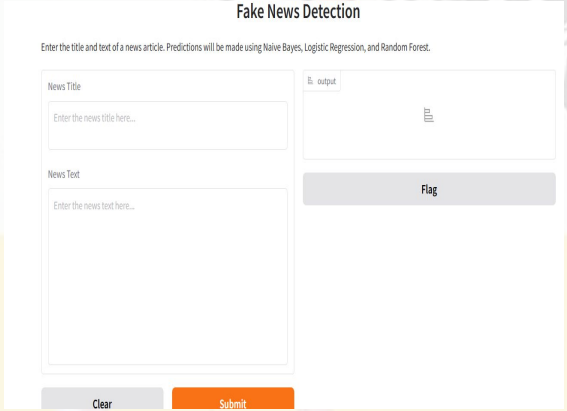
- F1-score average ≈ 0.88
- Important Words (LR): *trump, said, state, president*

Validation

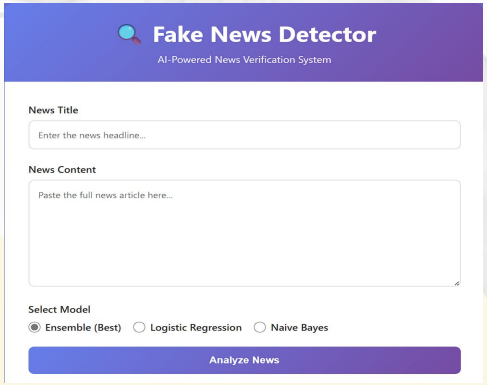
- It does not perform well (tendency seems correct)
- Presence of Leakage words

Deployment

Get it done easily



What we wanted… (docker + own server)
Spoiler!



Learning Takeaways

- **Data caution**: subjects affect real/fake likelihood and were considered in modeling.
- **Advanced preprocessing:** Handling emojis, contractions, corrections, and sentiment improves text understanding.
- **Embeddings:** Word embeddings capture semantic meaning and boost model performance beyond TF-IDF. Implementation?
- **Docker** deployment
- **Clean data** is essential to avoid overfitting.
- **Complexity** of the project from scratch
- **Time management**
- High level of **accuracy** doesn't mean the model is correct