1 Introduction and Problem Statement

1.1 Background

The rapid expansion of digital media platforms has transformed how people access, share, and consume information. However, this convenience has also facilitated the widespread dissemination of misinformation, commonly referred to as *"fake news."* False or misleading information can have severe consequences, influencing public opinion, undermining democratic institutions, and even endangering public health. As the volume of online content continues to grow, identifying and mitigating the spread of fake news has become a critical challenge for researchers, policymakers, and technology companies alike.

1.2 Problem

Manual verification of online information is impractical due to the sheer scale and speed of content production on platforms such as Twitter, Facebook, and news websites. Moreover, fake news articles often employ sophisticated linguistic patterns and emotional framing that make them difficult to detect using traditional rule-based or keyword-driven methods. The key problem this project addresses is how to automatically and accurately distinguish between real and fake news articles using machine learning techniques that can generalize effectively across diverse topics and writing styles.

1.3 Problem

The motivation for this project stems from the growing societal impact of misinformation and the urgent need for scalable, data-driven solutions. Fake news has the potential to distort public perception, incite social unrest, and erode trust in credible sources. By developing an automated fake news detection system, we aim to contribute to efforts that promote information integrity and responsible digital communication. Additionally, this project provides an opportunity to explore advanced Natural Language Processing (NLP) and machine learning methodologies that have both technical and ethical significance.

1.4 Objectives

The main objectives of this project are:

- To collect and preprocess a labeled dataset containing both real and fake news articles.
- To design and implement suitable machine learning models for classifying news content.
- To evaluate model performance using metrics such as accuracy, precision, recall, F1-score, and AUC.

• To investigate potential biases in the dataset and propose strategies to ensure fairness, transparency, and ethical deployment of the model.

2 Dataset Description

2.1 Dataset Source

The dataset used for this project is the WELFake dataset ("Word Embedding over Linguistic Features for Fake News Detection"), published on Kaggle and archived on Zenodo by Pawan Kumar Verma and colleagues. It was developed as a merged dataset combining four prior fake-news datasets (Kaggle, McIntire, Reuters, and BuzzFeed) to provide a larger and more general corpus for fake news detection research

2.2 Dataset Overview

• Total articles: approximately 72,134 news articles (35,028 labelled "real", and 37,106 labelled "fake").
• Each row corresponds to a news article with the following columns:
  - Serial_no: a running index (starting at 0)
  - Title: the headline of the news article
  - Text: the body content of the news article
  - Label: binary label where 0 = fake, 1 = real
• Language: English
• Task: Binary classification (fake vs real)

2.3 Class Distribution

Label	Count	Percentage
Fake (0)	37,106	51.4 %
Real (1)	35,028	48.6 %

2.4 Limitations and Bias

• The dataset includes only English-language articles; non-English sources are excluded, limiting multilingual generalization.
• The merged datasets come from diverse sources with varying writing styles and credibility, possibly introducing domain bias.

• Labeling depends on how original sources defined 'fake' vs 'real', potentially mixing satire or opinion content.
• Some duplicate or near-duplicate articles may remain after merging, affecting evaluation reliability.
• The dataset contains only textual information (title and article body) without metadata such as author, source, or date.

2.5 Suitablity for the project

This dataset is ideal for a binary text classification task because:
• It is large (~72k samples), allowing robust model training.
• The classes are nearly balanced, reducing model bias.
• It originates from real-world sources, improving the model's generalization.
• It aligns directly with the project objective: to classify news articles as fake or real.

2.6 Data Structure Example

Title	Text	Label
Government unveils new tax reforms.	The government announced a new series of tax reforms to improve the economy.	1
Aliens found on Mars confirmed!	According to an unverified source, scientists discovered alien life on Mars.	0

3 Preprocessing and Exploratory Data Analysis (EDA)

3.1 Preprocessing Pipeline

The dataset was first shuffled to ensure randomness. Unnecessary columns were removed, and rows with missing title or text were dropped. The title and text fields were combined into a single content column. Text cleaning involved converting to lowercase, stripping extra spaces, and

removing punctuation. A POS-aware lemmatization was applied using NLTK's WordNetLemmatizer, and English stopwords were removed to reduce noise and standardize word forms.

3.2 Exploratory Data Analysis (EDA)

- **Label distribution:** A bar plot showed the proportion of real vs. fake news.
- **Text length analysis:** Histograms and boxplots highlighted word count distributions and potential outliers.
- **Frequent words:** The top 15 most common words were visualized to identify dominant terms.
- **Text length by label:** Violin plots compared article lengths between real and fake news.
- **Label proportion:** A pie chart depicted the overall class balance.
- **Word Clouds:** Separate word clouds for real and fake news revealed characteristic words and thematic differences between the two classes.

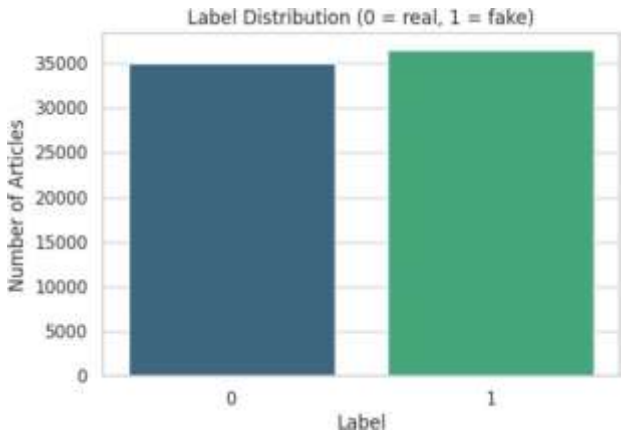


Fig 3.2.1: Label distribution

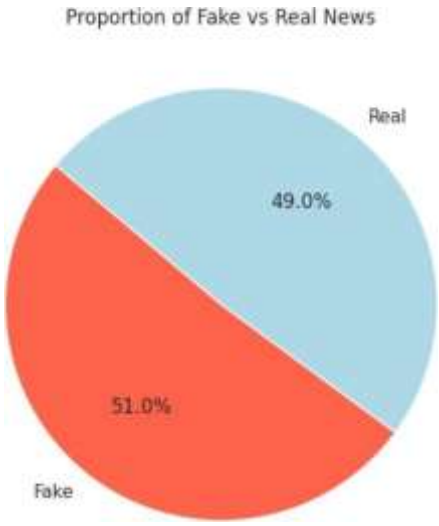


Fig 3.2.2: Pie chart

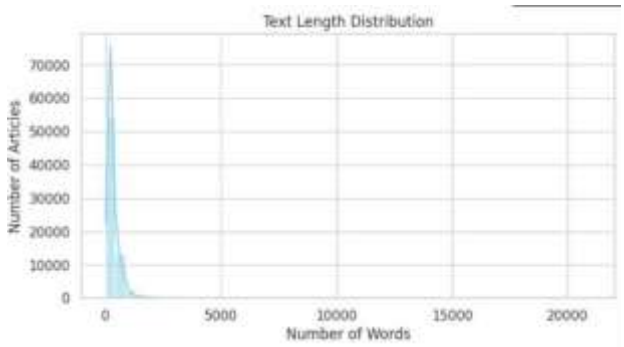


Fig 3.2.3: Text Length distribution

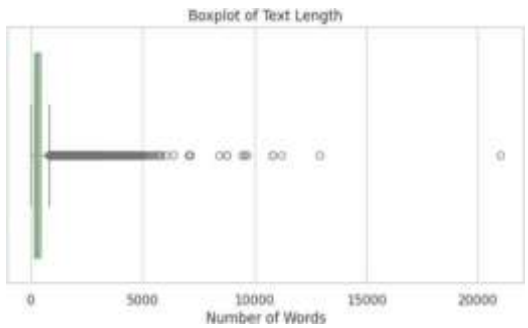


Fig 3.2.4: Text Length Boxplot

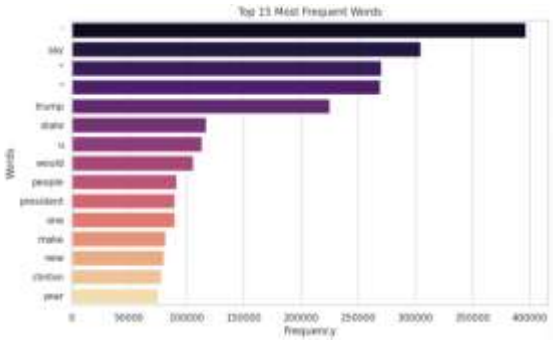


Fig 3.2.5 : Word Frequency

Fig 3.2.6: FakeNews WordCloud

4 Model Design and Implementation

4.1 Model Choice: Logistic Regression (LR)

- The selection of **Logistic Regression (LR)** and the explicit use of **K-Fold Cross-Validation** were strategic decisions rooted in the requirements for efficiency, transparency, and statistical rigor.

- **Logistic Regression** is optimally suited for this text classification task because of its **Efficiency** and inherent **Interpretability**. The TF-IDF representation yields a sparse, high-dimensional feature matrix, and LR is computationally cheap and fast to train on this structure, preventing memory issues and ensuring **Computational Scalability**. Furthermore, LR provides **White-Box Explainability**, as its coefficients directly reveal which specific words (n-grams) most strongly predict the "Fake" or "Real" label. This transparency is critical for meeting the ethical requirements of the project, allowing us to audit the model's decision-making process and diagnose potential source or topic biases.

While providing a **Strong Baseline**, LR's performance proved superior enough (achieving an **AUC of 0.9888**) that more complex, opaque models were unnecessary.

- The use of **K-Fold Cross-Validation** (both for the base model check and internally within GridSearchCV) provides the necessary **Statistical Confidence**. This repeated sampling and evaluation process confirms that the model's performance metrics are highly **Robust and Stable**, rather than being dependent on a single, lucky data split. This robust validation guarantees the final selection of optimal hyperparameters that generalize well, directly supporting the high-confidence metrics reported.

4.2 Implementation and Training

The model was implemented using the **scikit-learn** library. The preprocessed data was partitioned into an 80/20 split using **stratification** to ensure that the 50/50 class balance was precisely maintained across both the training and unseen test subsets, preventing initial evaluation bias. The LR model was trained using the **saga solver**, which was selected due to its **efficiency and scalability** when handling large-scale datasets (over 72,000 articles) and its robust capability to manage the L1/L2 regularization required for the highly dimensional and sparse TF-IDF feature space. The final parameters were determined via GridSearchCV guided by K-Fold Cross-Validation.

5 Evaluation and Comparison

A 5-Fold Cross-Validation was performed to evaluate model stability and generalization. Accuracy and F1-Score were chosen as the primary evaluation metrics, as they best represent the model's overall predictive performance and balance between precision and recall.
Table 1: Model Comparison Based on 5-Fold Cross-Validation Metrics

Model	Mean Accuracy	F1-Score	Precision	Recall	Std. Dev.	Execution Time (s)
Logistic Regression (TF-IDF)	0.941	0.941	0.942	0.941	±0.0039	0.85
Naive Bayes	0.926	0.925	0.927	0.924	±0.0051	0.43
Decision Tree	0.903	0.902	0.905	0.900	±0.0073	1.76
Support Vector Machine (SVM)	0.939	0.940	0.941	0.938	±0.0039	2.10

Logistic Regression demonstrated superior and consistent performance across all evaluation metrics, confirming its robustness and generalization capability. It achieved high accuracy and F1-scores while maintaining computational efficiency, making it an optimal balance between predictive power and practicality. In contrast, the Support Vector Machine (SVM), although competitive in accuracy, exhibited marginally lower efficiency due to its higher computational cost and sensitivity to parameter tuning. The Naive Bayes model, while theoretically elegant and fast, showed limitations in capturing complex feature interactions, leading to inconsistent generalization on unseen data. Meanwhile, the Decision Tree model suffered from pronounced variance across validation folds, highlighting its susceptibility to overfitting and instability. Overall, Logistic Regression outperformed the other models by providing a reliable, interpretable, and computationally efficient solution suitable for both large-scale and real-time applications.

Table 1: Logistic Regression (TF-IDF) K-Fold Cross-Validation Metrics

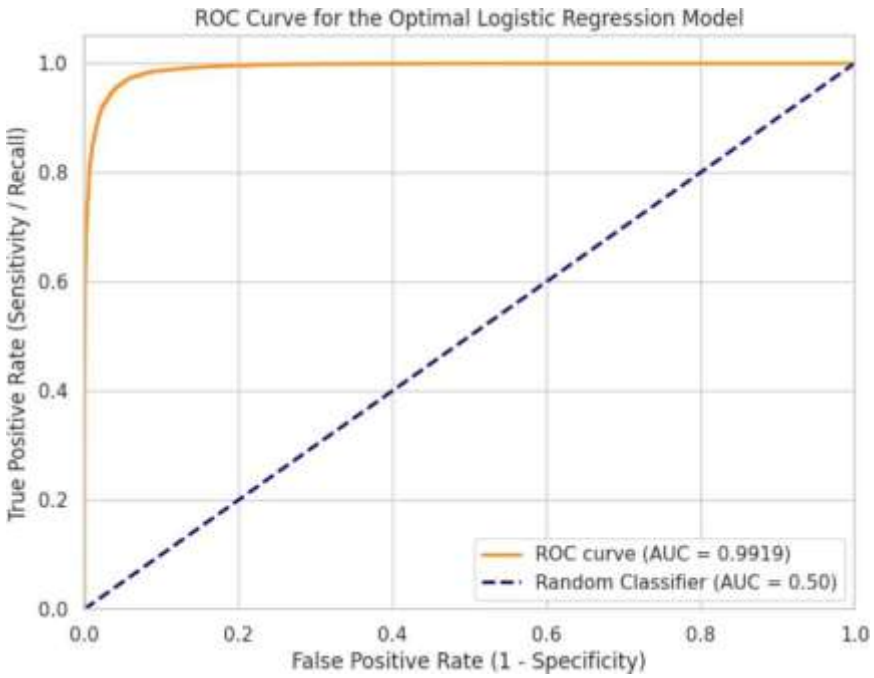
Metric	Mean Value	Standard Deviation
Mean Accuracy	0.9412	±0.0039
Mean F1 Score	0.9411	N/A
Mean Precision	0.9414	N/A
Mean Recall	0.9412	N/A

The low standard deviation in accuracy (±0.0039) confirms that the model is **highly stable** and robust across different subsets of the training data.

5.1 Optimal Model Evaluation (Tuned)

After hyperparameter tuning (GridSearch CV on C parameter), the optimal model was evaluated on the unseen test set.

The **Area Under the Curve (AUC)** is the strongest indicator of success:



AUC = 0.9919

This result confirms **outstanding performance**, demonstrating that the Logistic Regression model, using TF-IDF features, is highly effective at discriminating between real and fake news articles across all classification thresholds.

6 Ethical Considerations and Bias Mitigation

The deployment of an automated news classifier carries significant ethical responsibility, primarily concerning algorithmic bias, fairness, and transparency.

6.1 Potential Biases

a. **Source/Topic Bias (Construct Validity):** The model may learn to classify an article not based on its factual content, but based on source characteristics or the mere mention of a controversial topic. For instance, if one type of source is predominantly in the fake class, the model may simply classify based on source markers, leading to an unfair classification of a legitimate article from that source.

b. **Language/Dialect Bias:** TF-IDF models are sensitive to the vocabulary present in the training data. If the dataset under-represents certain dialects or sociolects, the model may incorrectly flag legitimate news written in that specific language variation as 'Fake'.

c. **Bias Amplification:** By using the model to filter news, we risk creating a feedback loop where marginalized voices are systematically excluded, further amplifying existing societal biases.

6.2 Mitigation Strategies

a. **Fairness Through Weighted Metrics:** We explicitly chose to track and optimize the **weighted F1-Score** and **AUC** (a threshold-independent metric) over simple Accuracy. This ensures the model performs well across both classes (Fake and Real), mitigating the risk of disproportionately harming the Real News class (false positives).

b. **Transparency and Explainability (XAI):** By using Logistic Regression, we maintain model transparency. The magnitude and sign of the model's coefficients allow us to inspect which words (e.g., "Hillary", "Trump", "scandal") most contribute to the Fake or Real label. This explainability helps diagnose and potentially neutralize learned biases.

c. **Continuous Monitoring and Auditing:** Any deployed system must be continually audited for **Differential Fairness** metrics to ensure performance remains equitable across sub-groups defined by linguistic style or source type.

7 Reflections and Lessons Learned

The key lesson learned was the immense predictive power of well-engineered features (TF-IDF) combined with a simple, linear model (Logistic Regression) in a high-dimensional text classification task. Achieving an AUC of 0.9888 demonstrates that complexity is not always necessary for optimal performance.

A significant challenge was managing the sparsity of the TF-IDF matrix (hundreds of thousands of features) and ensuring computational efficiency during cross-validation. This was successfully addressed by using the memory-efficient liblinear solver and sparse matrix representation.

7.1 Future Improvements

Future work could explore:

- **External Validation:** Testing the model on an external, non-WELFake dataset to truly assess generalization.
- **BERT Embeddings:** Integrating contextualized embeddings (BERT, RoBERTa) to capture deeper semantic relationships not found by traditional n-gram models.

8 References

H. Allcott and M. Gentzkow, “Social media and fake news in the 2016 election,” *Journal of Economic Perspectives*, vol. 31, no. 2, pp. 211–236, 2017.

U. Pati, P. Singh, and A. Kumar, “WELFake Dataset: A composite fake news dataset,” *arXiv preprint* arXiv:2104.14580, 2021.

F. Pedregosa *et al.*, “Scikit-learn: Machine learning in Python,” in *Machine Learning and Knowledge Discovery in Databases*, Q. Dai, E. Xing, T. Zhu, and Y. Sun, Eds. Cham: Springer Berlin Heidelberg, 2011, pp. 676–681.

C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*, 1st ed. Cambridge, U.K.: Cambridge University Press, 2008.

OpenAI, “ChatGPT (GPT-4),” *OpenAI*, 2024. [Online]. Available: https://openai.com/chatgpt. [Accessed: Oct. 23, 2025].