Capstone 3 Project Proposal

Fake News Detection with NLP and Generative AI

Prepared by: Mesfin Kebede
Date: September 25, 2025

Problem Identification

Problem Statement Formation

The rapid spread of misinformation online poses a significant threat to public opinion, decision-making, and societal stability, particularly in areas such as health, politics, and elections. This project aims to develop a text classification model using natural language processing (NLP) techniques to accurately distinguish between fake and real news articles, enabling automated detection and flagging of misleading content.

The project will be completed within 4 weeks (by October 15, 2025), aligning with the capstone schedule. Success will be defined quantitatively as achieving at least 90% accuracy, ≥85% precision and recall on the fake news class, and an F1-score ≥ 0.87 on held-out test data. Additionally, the model must generalize well to unseen data and demonstrate interpretability by highlighting influential words in classification decisions. A simple real-time demo (e.g., Streamlit app) will be delivered to showcase usability for end-users.

Context

In today's digital landscape, social media and online platforms amplify the spread of fake news at unprecedented speed. During events such as the COVID-19 pandemic or election cycles, false information has created public health risks, eroded trust in institutions, and influenced voter behavior. Media platforms and governments require automated solutions to complement human fact-checking, which is often too slow and resource-intensive. This project will apply NLP methods and compare baseline models with advanced transformers to demonstrate robust classification.

Criteria for Success

Quantitative Goals	Accuracy ≥ 90%; Precision and Recall ≥ 85% on fake news class; F1-score ≥ 0.87
Generalization	The model performs well on unseen test data
Usability	A simple demo (e.g., Streamlit app) for real-time classification
Interpretability	Provide explanations for predictions (e.g., highlighting influential words)
Deadline	Completed within 4 weeks (by October 15, 2025)

Scope of Solution Space

The project will focus on supervised learning approaches with text classification:

- Baseline: TF-IDF + Logistic Regression.
- Advanced: Fine-tuned Transformer (e.g., DistilBERT).
- Optional Extension: Use the Google Fact Check Tools API to benchmark predictions against verified fact-checks.

Out of scope: multimodal analysis (images/videos), full-scale production deployment, and multilingual systems.

Constraints

• Computational: Transformer fine-tuning requires GPU resources (Google Colab or Paperspace).
• Data Quality: Datasets may carry source bias, affecting fairness.
• Time: Accelerated capstone schedule limits deep hyperparameter tuning.
• Ethical: Care must be taken to avoid amplifying bias by unfairly labeling certain viewpoints as fake.

Stakeholders

• Primary: Media platforms (e.g., Google News, Twitter/X) who could integrate automated detection into moderation workflows.
• Secondary: Governments, fact-checking organizations (e.g., Snopes, Poynter Institute), journalists, and the public who rely on trustworthy information.

Data Sources

• Primary Dataset: ISOT Fake and Real News Dataset (~44,000 labeled articles).
• Optional Extension: Query Google's Fact Check API for cross-validation against verified fact-checks.

Approach Outline

1. Preprocessing & EDA: Clean text, tokenize, and explore data balance.
2. Modeling: Train TF-IDF + Logistic Regression baseline; fine-tune DistilBERT.
3. Explainability: Apply SHAP/LIME for interpretability.
4. Evaluation: Use accuracy, precision, recall, F1, and ROC-AUC.
5. Demo: Deploy a lightweight Streamlit app for real-time predictions.

Deliverables

• Code: Jupyter Notebooks with data wrangling, modeling, and evaluation.
• Metrics: Reported results (accuracy, precision, recall, F1, ROC-AUC).
• Report: 5–10 page technical writeup (problem, data, methods, results, insights).
• Slides: 10–15 slide deck for presentation.
• Repository: Organized GitHub repo with notebooks, report, slides, and model artifacts.
• Optional Demo: Streamlit web app for interactive predictions.