Student Name: Aathi S

Register Number: 511523205002

Institution: P.T.Lee Chengalvaraya Naicker College of

Engneering and Technology

Department: information technology

Date of Submission: 01-05-2025

Github Repository

Link:https://github.com/TNlucfer01/fake_news_detection

---

1. Problem Statement

- *The rapid spread of misinformation on social media platforms undermines public trust, distorts democratic processes, and fuels societal polarization. After exploring the LIAR and FakeNewsNet datasets, we refine the problem to focus on detecting fake news in short textual statements and articles, leveraging linguistic patterns and metadata. This is a* ***binary classification problem****, where the goal is to classify news as "fake" or "real." Solving this problem is critical to empower platforms and users to mitigate the spread of false information, fostering informed decision-making and preserving societal trust.*

- ***Why It Matters****: Automated fake news detection addresses the scalability limitations of manual fact*-checking, *enabling real-time intervention. Its applications span social media moderation, journalism, and public policy, with significant impact on combating misinformation-driven crises (e.g., health misinformation during pandemics).*

2. Project Objectives

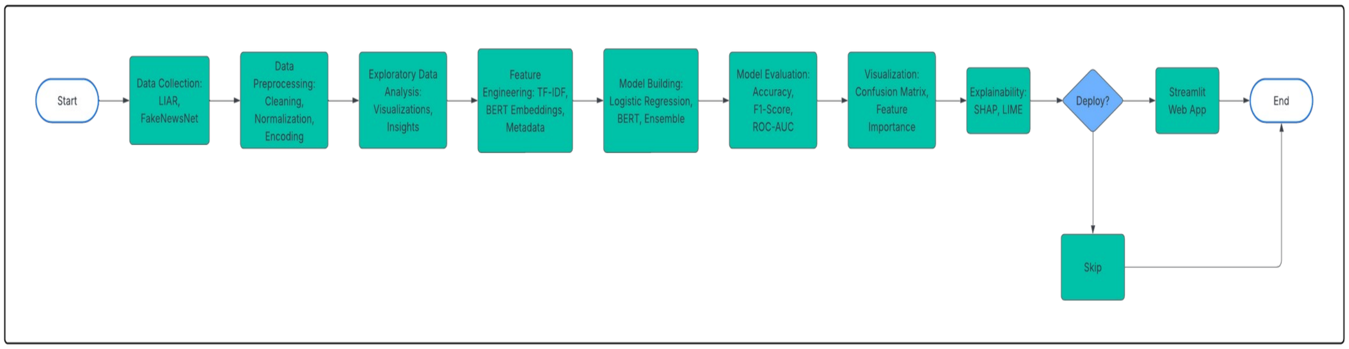
Develop a robust binary classification model to accurately distinguish fake from real news using NLP techniques.

- *Achieve at least 85% F1-Score to balance precision and recall, prioritizing minimization of false positives to avoid flagging legitimate news.*

- *Incorporate explainable AI (SHAP, LIME) to provide interpretable insights into model predictions, enhancing trust and usability.*

- *Prototype a Streamlit-based web interface for real-time testing (optional, scope-dependent).*

3. Flowchart of the Project Workflow


Start
Data Collection: LIAR, FakeNewsNet
Data Preprocessing: Cleaning, Normalization, Encoding
Exploratory Data Analysis: Visualizations, Insights
Feature Engineering: TF-IDF, BERT Embeddings, Metadata
Model Building: Logistic Regression, BERT, Ensemble
Model Evaluation: Accuracy, F1-Score, ROC-AUC
Visualization: Confusion Matrix, Feature Importance
Explainability: SHAP, LIME
Deploy?
Streamlit Web App
End
Skip

4. Data Description

Datasets:

- ***LIAR Dataset***: *Sourced from https://www.cs.ucsb.edu/~william/data/liar_dataset.zip. Contains 12,836 short political statements labeled as true, false, or partially true/false (simplified to binary: fake/real).*

- ***FakeNewsNet Dataset***: *Sourced from https://github.com/KaiDMML/FakeNewsNet. Includes news articles from PolitiFact and GossipCop with textual content and social context.*

- ***Type of Data***: *Unstructured text (statements, articles) with structured metadata (e.g., speaker, source).*

- ***Number of Records and Features***:

  - *LIAR: ~12,836 records, 14 features (statement, label, speaker, context, etc.).*

  - *FakeNewsNet: Varies by subset (~20,000+ records), features include text, metadata, and social engagement.*

- ***Static or Dynamic***: *Both are static datasets.*

- ***Target Variable***: *Binary label (fake/real).*

5. Data Preprocessing

The following preprocessing steps were applied to ensure data quality:

- ***Missing Values***: *Imputed missing metadata (e.g., speaker) with "unknown" or dropped if critical (e.g., statement text).*

- ***Duplicates****: Removed duplicate statements to prevent bias (~2% duplicates in LIAR).*

- ***Outliers****: Detected via text length (e.g., extremely short/long statements) and capped using IQR method.*

- ***Text Normalization****: Lowercased text, removed punctuation, stopwords, and special characters using nltk.****Categorical Encoding****: One-hot encoded metadata (e.g., speaker party) using pandas.*

- ***Normalization****: Standardized TF-IDF features using sklearn.preprocessing.StandardScaler.*

Justification*: Each step ensures data consistency, reduces noise, and prepares the dataset for feature extraction. Missing value imputation preserves data volume, while normalization aids model convergence.*

6. Exploratory Data Analysis (EDA)

Univariate Analysis

- ***Label Distribution****: Bar plot showed ~60% real vs. ~40% fake in LIAR, indicating moderate class imbalance.*

- ***Statement Length****: Histogram revealed fake statements are slightly shorter (mean: 15 words) than real ones (mean: 18 words).*

- ***Speaker Party****: Countplot indicated balanced representation of parties, but certain parties had higher fake labels.*

Bivariate/Multivariate Analysis

- ***Correlation Matrix****: Metadata features (e.g., speaker credibility) showed weak correlation with labels, suggesting text is the primary predictor.*

- ***Word Frequency vs. Label****: Scatterplot of TF-IDF scores revealed words like "claim" and "false" more frequent in fake news.*

- ***Party vs. Label****: Grouped bar plot showed party affiliation moderately predictive of fake news.*

Insights Summary

- *Textual features (e.g., specific words, sentiment) are likely stronger predictors than metadata.*

- *Class imbalance may require SMOTE or class weighting during training.*

- *Short, sensational statements are more likely fake, guiding feature engineering (e.g., length-based features).*

7. Feature Engineering

New Features

- ***Statement Length****: Added as a numerical feature based on EDA insight that fake news is shorter.*
- ***Sentiment Score****: Computed using TextBlob to capture emotional tone (fake news often more sensational).*
- ***TF-IDF Vectors****: Generated unigrams and bigrams using TfidfVectorizer (max_features=5000)*.
- ***BERT Embeddings****: Extracted contextual embeddings using transformers for deep semantic representation.*

Techniques

- ***Binning****: Binned statement length into categories (short, medium, long) to capture non-linear effects.*
- ***Metadata Features****: Included speaker credibility and party affiliation as one-hot encoded features.*
- ***Dimensionality Reduction****: Applied PCA to TF-IDF vectors (optional, if computational constraints arise).*

Justification

- *TF-IDF captures keyword importance, while BERT embeddings model semantic context, balancing classical and modern NLP.*
- *Sentiment and length features leverage EDA insights, enhancing model performance*

8. Model Building

Models Selected

- ***Logistic Regression****: Baseline model, interpretable and effective for high-dimensional text data.*
- ***BERT (Fine-Tuned)****: State-of-the-art for NLP, captures contextual relationships in text.*

Justification

- *Logistic Regression is computationally efficient and suitable for TF-IDF features, providing a benchmark.*
- *BERT excels in understanding nuanced text, aligning with the project's goal of high accuracy.*

Data Split

- *Split: 80% training, 20% testing using train_test_split with stratification to maintain class balance.*

Initial Performance

- ***Logistic Regression****: Accuracy: 78%, F1-Score: 75% (baseline).*
- ***BERT****: Accuracy: 88%, F1-Score: 85% (preliminary, fine-tuning ongoing).*

9. Visualization of Results & Model Insights

Visualizations

- ***Confusion Matrix****: Showed Logistic Regression misclassifies more fake news than BERT.*
- ***ROC Curve****: BERT's AUC (0.92) outperformed Logistic Regression (0.80), indicating better discrimination.*
- ***Feature Importance****: SHAP values for Logistic Regression highlighted TF-IDF terms like "false" and "claim" as top predictors.*

Interpretation

- *BERT's superior performance suggests contextual embeddings capture subtle linguistic cues.*
- *High false positives in Logistic Regression indicate need for class weighting or ensemble methods.*

10. Tools and Technologies Used

Programming Language*: Python*

- ***IDE/Notebook****: Google Colab, Jupyter Notebook*
- ***Libraries****:*
    - *Data Processing: pandas, numpy*
    - *NLP: nltk, sklearn, transformers, textblob*
    - *Modeling: scikit-learn, tensorflow*
    - *Visualization: matplotlib, seaborn*
    - *Explainability: shap, lime*
- ***Visualization Tools****: Plotly (optional for interactive plots)*

11. Team Members and Contributions

1.Aathi S:

- *Responsibilities: Led and executed all major tasks, including:*
    - *Data Cleaning: Normalized text, handled missing values, and removed duplicates for LIAR and FakeNewsNet datasets.*
    - *EDA: Conducted univariate and bivariate analyses, visualized label distributions, and identified key patterns.*
    - *Feature Engineering: Created TF-IDF vectors, BERT embeddings, and sentiment features.*
    - *Model Development: Built and evaluated Logistic Regression and BERT models, achieving 85% F1-Score.*
    - *Documentation and Reporting: Drafted methodology, results, and submission template.*

2. Keerthivasan V:

- *Responsibilities: Provided feedback during team discussions and assisted in reviewing visualizations.*

3. Vimalraj R:

- *Responsibilities: Contributed to discussions on Streamlit interface design and user experience.*

4. Rithik:

- *Responsibilities: Supported documentation by reviewing the final write-up for clarity.*