CIS 4190/5190: Applied Machine Learning Fall 2025

Binary News Source Classification from URL Slugs

Team Members: Amogh Channashetti, Mohit Singh, Yohan Vinu *Project: News Source*

News Classifier 1M URLs Dataset

1 Introduction

This project builds a binary news-source classifier that distinguishes between Fox and NBC articles using only lightweight signals, namely headlines derived from URL slugs. We experimented with multiple models, including logistic regression, transformer models (DistilBERT-base, RoBERTa-base, and DistilBERT-small), random forests, XGBoost, and an LSTM.

One of our strongest models uses a logistic regression classifier on TF–IDF character n-grams extracted from URL slugs. We carefully preprocessed URLs to infer labels from domains, normalize paths, remove tracking-style ID segments, and retain only informative slug tokens. A small hyperparameter sweep over the regularization strength C was performed on a held-out validation split. The best model was then retrained on the full dataset and packaged, including both the vectorizer and classifier, into a compact artifact for downstream use.

Our second pipeline fine-tunes a DistilBERT sequence classifier on raw headline text from a labeled CSV file. We applied standard data cleaning steps such as dropping missing labels or headlines, restricting to valid classes, and using a stratified 80/20 train–validation split. Training was conducted using the HuggingFace Trainer API with appropriate hyperparameters, including batch size, warmup steps, weight decay, per-epoch evaluation, and best-model checkpointing. Performance was evaluated using accuracy and macro-F1. Both pipelines substantially outperformed a majority-class baseline, confirming that partisan signal is strongly encoded even in surface-level textual features.

Beyond core modeling, we conducted exploratory data analysis on the headline corpus to understand stylistic and linguistic differences between sources. This analysis included class balance checks; length and complexity measurements such as word and character counts and Flesch–Kincaid readability; clickbait indicators including exclamation marks, question marks, and all-caps ratios; distinctive vocabulary analysis using log-odds to identify source-specific terms; sentiment analysis with VADER; and named entity and readability profiling. These analyses revealed systematic differences in vocabulary usage, sentiment distributions, and readability levels between Fox and NBC headlines. Together, they provide qualitative evidence that complements the quantitative model results and helps explain why even relatively simple models can reliably distinguish between the two outlets.

2 Core Components

2.1 Data Collection and Dataset

To ensure consistency, we used URL-derived titles as the primary text source and applied standard text cleaning techniques, including normalization and lemmatization. URLs that lacked descriptive content (such as those consisting primarily of non-semantic path segments) were identified and removed, as they did not provide meaningful signals for classification. Approximately 20% of collected URLs were excluded through this filtering process, resulting in a final dataset of roughly 770,000 usable headlines suitable for training and evaluation.

2.2 Model Design

We began by establishing a baseline using a Logistic Regression model trained on TF-IDF vectors using character n-grams (3–6 characters). This simple approach yielded a strong validation F1 score of 86.51%.

To explore non-linear capabilities, we attempted to use these same sparse TF-IDF representations with tree-based models. However, as expected, these architectures struggled with the high dimensionality and sparsity of the feature space; Random Forest achieved an F1 score of 71%, and XGBoost achieved 69%.

Building on these findings, we hypothesized that deep contextual embeddings were necessary to break the performance ceiling. We expanded our dataset to 1 million URLs and transitioned to Transformer-based architectures. We experimented with RoBERTa Base and DistilBERT to leverage their pre-trained understanding of sub-word tokens. Empirically, DistilBERT (base-uncased) proved to be the optimal architecture, achieving an F1 score of 89.9%, notably outperforming the heavier RoBERTa model (87%). We attribute this to the fact that URL classification relies less on deep, complex reasoning, where RoBERTa typically excels, and more on identifying short-range semantic associations, for which DistilBERT's distilled architecture is sufficient and less prone to overfitting.

Finally, we conducted an ablation study to test if a recurrent architecture could offer a lightweight alternative to Transformers. Since URLs lack the stopwords and complex grammatical structure of natural language, we hypothesized that an LSTM could capture the necessary sequential dependencies without the computational overhead of Self-Attention mechanisms. While the LSTM achieved a respectable 86% accuracy, it failed to surpass the Logistic Regression baseline or the Transformer models, leading us to finalize DistilBERT as our production architecture for its superior balance of precision and recall.

2.3 Evaluation and Model Performance

We evaluated model iterations using an 80/20 train–test split on the preprocessed dataset to assess generalization performance prior to submission. All models were trained on the training split and evaluated on a held-out test set drawn from the same distribution, allowing for consistent comparison across iterations. For model development on smaller datasets (e.g., the provided course dataset), we additionally used cross-validation to maximize data utilization and reduce variance in performance estimates. For large-scale training and final evaluation on our curated dataset, we used fixed, stratified train/validation/test splits to ensure comparability across models and reproducibility of results.

Model performance was primarily measured using classification accuracy, consistent with the evaluation metric used by the course baseline and leaderboard. Additional metrics such as precision, recall, and F1-score were examined to better understand class-level behavior and ensure balanced performance across news sources.

Models selected for leaderboard submission were chosen based on their performance on the internal test set, prioritizing those that demonstrated the strongest results across metrics. This approach ensured that submitted models reflected improvements rather than overfitting to the training data.

3 Exploratory Components

3.1 Code

To address the constraints of our limited 4,000-sample dataset, our primary motivation was to maximize data utility and ensure statistical robustness. We conducted initial development and cross-validation on the provided ∼4,000-sample course dataset, and separately trained large-scale models on our curated 770K–1M URL dataset for extended analysis. We implemented a 4-fold Cross-Validation strategy rather than a simple train-test split, allowing us to train and validate on the entire dataset to reduce overfitting risks. To further improve efficiency, we focused on vectorizing the training loop; by batching all operations, we maximized GPU saturation and minimized the overhead associated with sequential processing.

Our development approach relied on a hybrid hardware workflow to balance computational power with rapid prototyping. We utilized an NVIDIA A100 GPU via Google Colab for production-grade training runs where high-bandwidth memory was essential. Conversely, to preserve compute units and accelerate the debugging cycle, we conducted initial dry-runs and syntax validation locally on an RTX 5070 using CUDA. We managed this distributed environment through GitHub, maintaining four distinct branches to isolate experimental setups, which ensured that testing new methodologies did not destabilize the main training pipeline.

The impact of these contributions was evident in both development velocity and model reliability. The transition to batched operations on the A100 yielded a significant speedup in epoch processing time compared to our unoptimized baseline. Furthermore, the 4-fold validation strategy revealed a more consistent F1-score across folds, confirming that our model's performance was a result of learned features rather than data variance. This rigorous testing setup allowed us to confidently deploy the final model with verified generalization capabilities.

3.2 Technique

To address the significant class imbalance inherent in our dataset, we adopted a Cost-Sensitive Learning approach via a Weighted Cross-Entropy Loss function. Our primary motivation was to maximize data utility; standard mitigation techniques, such as random undersampling of the majority class, would have necessitated discarding valid data points. Given we lost 230,000 samples from NBC due to invalid URLs (e.g: nbc/wnba324987.com), removing data was deemed suboptimal. By theoretically grounding our approach in cost-sensitive learning, we ensured that the model was penalized disproportionately for misclassifying the minority class, effectively forcing the optimization landscape to respect the distribution of the rarer class without any loss of training information.

We integrated this technique directly into the training pipeline by modifying the loss criterion rather than the optimization algorithm itself. We calculated scalar weights for each class using the Inverse Class Frequency method, where the weight W_c for a class is inversely proportional to its prevalence in the training set ($W_c \approx \frac{N_{total}}{N_{classes} \cdot N_c}$). These weights were then passed as the `weight` argument to the `CrossEntropyLoss` function. Consequently, while we utilized the standard AdamW optimizer, the gradients propagated during backpropagation were scaled by these class-specific weights, steering the model updates to prioritize the minority class.

The validation of this technique involved a direct comparison against a baseline model trained with standard, unweighted Cross-Entropy Loss. Our ablation studies demonstrated that while the unweighted baseline achieved high overall accuracy, it suffered from poor recall on the minority class, effectively biasing predictions toward the majority. The introduction of the weighted loss resulted in a marked improvement in the minority class F1-score and Recall, confirming that penalizing specific error types is a more effective strategy for this dataset than altering the underlying data distribution.

3.3 Analysis

Research Question and Hypothesis
This project asks whether the partisan source of a news article (Fox vs. NBC) can be inferred using only lightweight textual signals, specifically URL-derived headlines, and what systematic linguistic or stylistic differences enable this separability. We hypothesized that even short, surface-level text encodes sufficient framing and vocabulary cues for classifiers to perform well above chance and majority baselines, with these differences observable through exploratory analysis.

Experimental Design and Methods
We constructed a labeled dataset of 771,410 headlines (65% Fox, 35% NBC), inferred from article domains and split into 80/10/10 train/validation/test sets using stratified sampling. In parallel, we built a URL-only pipeline that cleaned and tokenized URL slugs by removing boilerplate, IDs, and uninformative tokens.

We evaluated a range of models: logistic regression with TF–IDF character n-grams, tree-based models, an LSTM, and fine-tuned transformers (DistilBERT and RoBERTa). Logistic regression hyperparameters were tuned on validation data, while transformers were trained with class-weighted loss, mixed precision, and early stopping. Performance was measured using accuracy and macro-F1 to account for class imbalance.

To contextualize model behavior, we conducted exploratory analyses on headline length, punctuation and capitalization (clickbait proxies), sentiment (VADER), readability, distinctive vocabulary via smoothed log-odds, and named entity frequency.

Primary Findings

All models exceeded the 66% majority baseline, confirming strong partisan signal in headlines and URL slugs. DistilBERT achieved 0.875 accuracy and 0.865 macro-F1, with balanced performance across classes; RoBERTa showed comparable results and strong generalization. The TF–IDF logistic regression model performed competitively despite its simplicity, demonstrating that shallow textual cues alone are highly informative.

EDA supported these results: Fox and NBC differed systematically in vocabulary, sentiment distributions, readability levels, punctuation usage, and entity emphasis. Altogether, these patterns explain why even lightweight models can reliably distinguish sources.

Insights and Limitations

The key insight is that source identity is highly recoverable from minimal text, highlighting the strength of stylistic and lexical cues in partisan media. However, labels are domain-based and noisy, the study is limited to two outlets, and performance may partially reflect outlet-specific formatting artifacts rather than ideology alone. As a result, such models are best used as analytical or auditing tools and should be complemented with broader datasets and qualitative analysis.

3.4 Datasets

News Classifier 1M URLs Dataset

The motivation for collecting a new dataset came from the limitations of the data provided for the news source classification task. The baseline dataset was useful for initial experimentation, but its relatively small size constrained the ability to explore more robust models and limited the evaluation of generalization to unseen data. To address this, we curated a larger dataset of news headlines to better reflect real-world distributions and build more reliable models.

The dataset was sourced by scraping news article headlines from FOX News and MSNBC. URLs were collected programmatically and used as inputs to an automated scraping pipeline that extracted the headline text from article pages. We employed a BFS-style approach to ensure that we don't skew our training set towards specific subdomains, such as "fox/sports/. . . ". Source labels were inferred deterministically based on the originating domain, ensuring a reproducible labeling process.

To maintain consistency across the dataset, only articles from these two outlets were included, and each headline was associated with a single source label. Any entries that could not be confidently attributed to either source were explicitly removed, as well as any duplicates we encountered. URLs specifically for NBC we encountered were often formatted without a title, so we removed them from our training set as they were essentially invalid.

The URLs underwent a series of cleaning and normalization steps to ensure data quality and usability. A key preprocessing step was lemmatization, which reduced words to their canonical forms in order to minimize vocabulary sparsity and reduce stylistic variation between sources. In some of our lightweight models that like as TF-IDF Logistic Regression, we also removed stop words; however, when we moved on to transformer models, we kept stop words to enable the attention layer to draw context from the relation between words. Also, we decoded any url special characters like %20 = "space" character.

Total URLs Crawled	1,000,200	
Duplicates Dropped	10,000	
Invalid URLs Removed	230,000	
Statistic	**Fox News**	**NBC News**
Sample Count	496,863	261,410
Average Word Count	9.419	7.901
Average Character Count	58.623	51.246
Flesch-Kincaid Readability Score	3.625	3.206
Average Sentiment Score	-0.111	-0.097

Table 1: Dataset statistics after cleaning and preprocessing.

4 Team Contributions

All team members contributed equally across all aspects of the project. Amogh primarily focused on the data collection and preprocessing pipeline, including scraping, cleaning, and curating the final dataset. Mohit led model exploration and experimentation, evaluating different modeling approaches and configurations. Yohan contributed to model training, evaluation, and analysis, including assessing performance across validation splits and interpreting results. All team members collaborated on writing and refining the final project report.

5 Extra Credit

Video Link: https://drive.google.com/file/d/1mkjHgbsLR87UP192y9EjL6CxSYH2utPB/view?usp=sharing