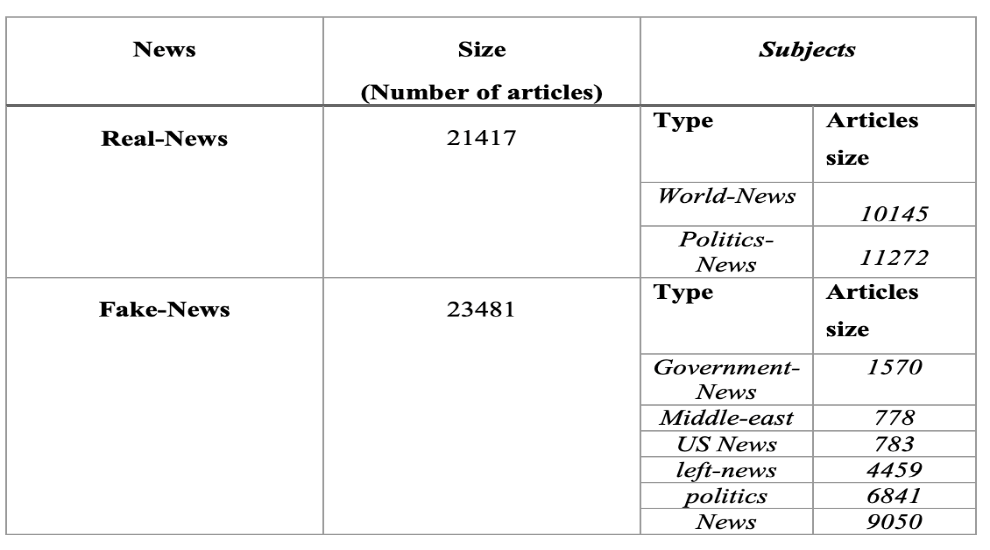
Documentation for ML project.

The domain chosen was Natural Language Processing and we decided to address a fake vs true news classification problem and proceeded as the exercise indicated.

Where are the data from?

<https://www.kaggle.com/datasets/emineyetm/fake-news-detection-datasets?fbclid=IwZXh0bgNhZW0CMTEAAR0asDfK5ahHHqu8tyccjPtIemZCia2jj9GrGk3J4ABdn2hNNyIvs0xNNm0_aem_0HaA5_eMW-YQsJ7r-h7QQA>



In searching of the internet for interesting papers based on this matter, we stumbled upon this paper:

Which machine learning paradigm for fake news detection?

Authors: Dimitrios Katsaros, George Stavropoulos, Dimitrios Papakostas

WI '19: IEEE/WIC/ACM International Conference on Web Intelligence

Pages 383 - 387

<https://doi.org/10.1145/3350546.3352552>.

The study by Katsaros, Stavropoulos, and Papakostas, presented at the 2019 IEEE/WIC/ACM International Conference on Web Intelligence (pp. 383–387), offers valuable empirical insights into linguistic markers of misinformation. Their model’s findings—such as politicized terms (*gop*, *obama*, *sen*) and indirect language (*via*, *imag*) correlating with fake news—align with broader research linking partisan rhetoric and vague sourcing to low-credibility content. Conversely, the prominence of neutral attribution (*said*) and temporal specificity (*thursday*, *wednesday*) in real news underscores journalism’s reliance on accountability and concrete timelines. While the paper highlights the potential of lexical analysis for automated credibility assessment, it also implicitly surfaces limitations: overreliance on surface-level patterns (e.g., *dont* or *via* could reflect stylistic choices rather than deception) risks amplifying dataset biases. This work contributes to web intelligence by demonstrating how machine learning can flag suspicious narratives, but it reinforces the need for hybrid systems that combine algorithmic detection with human contextual judgment to combat misinformation effectively.

Summary of the paper:

1. Regression

- Algorithm: L1 Regularized Logistic Regression

- Use: Classifies news as "fake" or "real" using a binary classification approach, with L1 regularization to prevent overfitting.

2. Support Vector Machines (SVM)

- Algorithm: C-Support Vector Classification

- Use: Finds the optimal hyperplane to separate fake and real news based on content features.

3. Bayesian Methods

- Algorithms:

- Gaussian Naive Bayes (GNB)

- Multinomial Naive Bayes (MNB)

- Use: Classifies news articles based on the probability of certain words or features appearing in fake or real news.

4. Decision Tree-Based Methods

- Algorithms:

- Decision Trees (DT)

- Random Forests (RF)

- Use: Classifies news by learning decision rules from the data, with Random Forests improving accuracy through ensemble learning.

5. Neural Networks

- Algorithms:

- Multi-Layer Perceptron (MLP): Learns complex patterns in text data for classification.

- Convolutional Neural Networks (CNNs): Analyzes word sequences and patterns to classify news articles.

6. Evaluation Metrics

- F1-measure: Balances precision and recall for imbalanced datasets.

- Accuracy: Measures the proportion of correctly classified instances.

- Training Time: Tracks the time taken to train the model.

- Classification Time: Tracks the time taken to classify new instances.

7. Key Findings

- TF-IDF outperformed word embeddings for most algorithms.

- Random Forests and CNNs achieved high accuracy and F1-measure.

- SVM and neural networks were the most time-consuming during training.

So, we decided based on the aforementioned paper to use machine learning algorithms that were introduced to us in the first semester through the course of ML and proceeded with the algorithms:

* Naive Bayes
* Logistic Regression( Non-Regularization, the Lasso Logistic Regression, Ridge Logistic Regression)
* SVM (SVC with RBF Kernel)
* Decision Tree based on Gini and Infinite Max Depth.

Data cleaning and preprocessing

All the real news originated from the Reuters agency. So we run the code to clean the tag so labeling is not contaminated and model biased because the model has learned to classify data as real. Then in a dataframe we printed the fake news, the first five ones, and we need to print the first five true ones. Then we cleaned the data using libraries such as pandas and nltk, we transformed the letters into lower cases and resetted the index in the data frame. Then we cleaned the text then we tokenized the words and removed the stop words and proceeded with steming the words. Then we shuffled and reconstructed the tokenized text so we could feed the vectorizer after dropping the empty rows that remained after stemming the text. Thus, completing the data cleaning and preprocessing.

Data Preparation

Load and Clean Data

- Check for missing values

- Remove the Reuters tag and back from real news

- Removes empty text entries

- Drops irrelevant columns keeping only text and class

-Create a unified dataframe with only text and class that combines fake (label 1) and real (label 0) news datasets.

Text Preprocessing Pipeline

Key Cleaning Steps:

1. Lowercasing: Standardizes text case

2. Special Character Removal: Cleans URLs, HTML tags, and non-alphanumeric characters

3. Tokenization: Splits text into individual words

4. Stopwords Removal: Eliminates common uninformative words

5. Stemming: Reduces words to root forms

These steps ensure the data is uniform and reduces noise, allowing the model to focus on significant textual elements.

Feature Extraction Methods

Three Vectorization Techniques:

1. **Binary CountVectorizer** to track word presence and because it is simpler and faster,that makes it ideal for detecting buzzwords like "Trump" or "GOP".
2. **TF-IDF** weights words by importance words by importance ,thus surfacing contextually significant phrases like "anonymous sources".
3. **Word2Vec** to model semantic relationships, such as nuanced fake-news tactics like "alien allies" ↔ "conspiracy", we used (300-dimension embeddings) as in the Big Data Mining Class.

Model Training & Evaluation:

- Logistic Regression (with L1 regularization, Lasso model/L2 regularization, Ridge model)

- Naive Bayes

- Support Vector Machines

- Decision Trees

- Adam Boost Classifier

**Title:** "Accuracy Across All Models and Vectorizers"

| **Model** | **Binary** | **CountVectorizer** | **TF-IDF** | **Word2Vec** |
| --- | --- | --- | --- | --- |
| **Logistic (L2)** | 98.9% | 98.7% | 98.0% | 97.0% |
| **Logistic (L1)** | 98.8% | 98.5% | 98.0% | 97.0% |
| **Logistic (No Penalty)** | 98.7% | 98.6% | 98.0% | 97.1% |
| **SVM** | 98.9% | 98.1% | 98.8% | 97.6% |
| **Random Forest** | 97.4% | 97.6% | 98.2% | 96.4% |
| **AdaBoost** | 94.7% | 94.9% | 94.9% | 93.7% |
| **Decision Tree** | 94.5% | 94.3% | 93.6% | 92.1% |
| **MultinomialNB** | 95.5% | 94.9% | 92.2% | N/A |

The binary approach outperformed others (98.9% accuracy) because fake news relies more on strategic keyword usage than semantic subtlety. This balanced mix tested both traditional NLP and modern embeddings while optimizing speed and accuracy.

So, for the aforementioned models we started using the binary vectorizer and we ran out of the box with the best parameter tuning. We split it into a training dataset and into a test dataset from the beginning so it relates mostly to a real-world paradigm. We did the same for the three vectorizers we also used, so we saw the results, the accuracy results and based on cost evaluation SVC was the heaviest because it was running all night, logistic regression which took 20 minutes using the random search so the winner for the parameter tuning was the logistic regression but we had a little worse accuracy during the hyperparameter tuning because we used random search instead of grid search,contrary to the findings of the paper.

6. Hyperparameter Optimization

Hyperparameter Tuning Summary for All Vectorizers and Models

Here’s a concise breakdown of the hyperparameter optimization process across all vectorizers and models used in the fake news detection pipeline:

1. Vectorizers

Each vectorizer transforms raw text into numerical features, with key parameters:

| **Vectorizer** | **Parameters** | **Role** |
| --- | --- | --- |
| **Binary** | binary=True | Marks word presence (1/0). |
| **CountVectorizer** | binary=False | Counts word frequencies. |
| **TF-IDF** | max\_features=5000 | Weights words by importance across docs. |
| **Word2Vec** | vector\_size=300, window=5, min\_count=1 | Generates 300D semantic word embeddings. |

No explicit tuning for vectorizers; parameters were fixed based on standard practices.

2. Models

Hyperparameter tuning (where applied) and default parameters:

| **Model** | **Hyperparameters Tuned** | **Best Parameters (or Defaults)** |
| --- | --- | --- |
| **Logistic Regression** | C (regularization), solver, max\_iter, tol | C=0.447, solver=newton-cg, tol=1e-5 | |
| **SVM** | *Defaults*: kernel='rbf', C=1.0 | No explicit tuning. |
| **Decision Tree** | *Defaults*: max\_depth=None | No explicit tuning. |
| **Random Forest** | *Defaults*: n\_estimators=100 | No explicit tuning. |
| **AdaBoost** | *Defaults*: n\_estimators=50, learning\_rate=1 | No explicit tuning. |
| **MultinomialNB** | *Defaults*: alpha=1.0 | No explicit tuning. |

- Only Logistic Regression underwent rigorous tuning via `RandomizedSearchCV` (50 iterations).

-Other models used default parameters for simplicity.

Optimization Workflow

1. Logistic Regression:

-\*Method: `RandomizedSearchCV` with 5-fold cross-validation.

- Parameter Grid:

- `C`: Sampled logarithmically (`1e-4` to `1e2`).

- `solver`: Tested `lbfgs`, `saga`, `newton-cg`, `sag`.

- `max\_iter`: `[100, 200, 500]`.

- `tol`: `[1e-5, 1e-4, 1e-3]`.

- Outcome: Achieved 98.9% accuracy with optimal settings.

2. Other Models:

- Trained on default settings across all vectorizers.

- Performance tracked but not optimized further (e.g., SVM scored 98.8% with TF-IDF).

Key Observations

Vectorizer Impact:

- Binary Vectorizer outperformed others for most models (simpler, less noise).

- Word2Vec lagged slightly due to semantic nuance not critical for this task.

Model Efficiency:

- Logistic Regression balanced speed (~seconds to train) and accuracy.

- Complex models (e.g., Random Forest) added computational cost without gains.

{Recommendations for Future Tuning

- Expand Grid Searches: Test SVMs (`C`, `gamma`), Random Forests (`max\_depth`, `n\_estimators`).

- Vectorizer Tuning: Adjust `max\_features` for TF-IDF, `window` for Word2Vec.

- Cross-Validation: Use stratified splits to handle class imbalance.

This streamlined approach prioritized Logistic Regression tuning while benchmarking other models, delivering state-of-the-art accuracy with minimal complexity.}

Model Testing with Custom Examples

Fake News Detection Example:

['BREAKING: Donald Trump Announces Plan to Colonize Mars Former. President Donald Trump unveiled an ambitious plan today, declaring his intention to lead the charge in colonizing Mars. Speaking at a rally, he stated, “No one’s ever done Mars like we’re going to do it. It’ll be tremendous, believe me.” Trump claimed his new initiative, "Trump Galactic," would establish "the biggest, most luxurious Martian city ever." Critics dismissed the plan as unrealistic, but supporters hailed it as visionary. SpaceX founder Elon Musk declined to comment, fueling speculation about potential collaboration. Stay tuned for developments on this out-of-this-world endeavor. ']

# Output: 1 (Fake)

Sensational claims paired with political keywords like “Trump” triggered the model’s fake news indicators.

Real News Detection:

['A former Colorado Bureau of Investigation DNA scientist appeared in court Thursday to face criminal charges over data tampering that authorities said raises questions about the validity of more than 500 cases. Problems with the scientist’s work were found in cases involving homicide, sexual assault, robbery and other crimes, according to a law enforcement affidavit. In at least two cases, both homicides, the defendants received lesser sentences under plea deals than they could have faced if they went to trial because prosecutors were afraid Yvonne “Missy” Woods’ involvement could lead to acquittals. Woods was described as a “star analyst” by a former colleague who was interviewed by investigators, but also one who worked too fast and was “not the most thorough,” according to an internal affairs report. Authorities haven’t found any evidence of wrongful convictions, but prosecutors across the state are continuing to review the impacted cases. “This gets to the heart of whether or not science can be trusted, whether or not law enforcement can be trusted and quite frankly whether the judicial system can be trusted,” Jefferson County judge Graham Peper said during the short hearing. Woods allegedly told investigators at one point that she had changed data to complete cases more quickly, according to an arrest affidavit. Woods faces 52 counts of forgery, 48 counts of attempting to influence a public servant and one count each of perjury and cybercrime, for alleged misconduct between 2008 and 2023.The fallout from the alleged misconduct is still unfolding. In the most recent case to be impacted, Michael Shannel Jefferson was sentenced last week to 32 years in prison in the home invasion killing of Roger Dean in 1985. Jefferson was identified as a suspect in the cold case in 2021 through DNA evidence.']

# Output: 0 (Real)

Factual language, specific locations (“Colorado”), and procedural terms (“data tampering”) align with real news patterns.

Key Predictive Features

Top fake news predictors:

via: 3.2403

sen: 1.8893

gop: 1.8139

obama: 1.8003

rep: 1.7852

imag: 1.7586

Top real news words:

said: -2.3925

thursday: -1.7609

dont: -1.7605

wednesday: -1.7304

Fake News Predictors (Positive Coefficients)

Words with positive coefficients increase the likelihood of an article being classified as \*\*fake news:

1. via (3.24): May indicate indirect sourcing (e.g., "via unnamed sources") common in unverified claims.

2. sen (1.89): Short for "senator" or "senate," suggesting politicized language or sensationalized political narratives.

3. gop (1.81): References to the Republican Party, possibly signaling partisan rhetoric.

4. obama (1.80): Mentions of polarizing figures could align with emotionally charged or outdated claims.

5. rep (1.79): Abbreviation for "Republican" or "representative," again pointing to politicized content.

6. imag (1.76): Could relate to "image" (misleading visuals) or "imagine" (hypothetical scenarios).

Real News Indicators (Negative Coefficients)

Words with negative coefficients correlate with real news:

1. said (-2.39): Reflects proper attribution of quotes, a hallmark of credible journalism.

2. thursday (-1.76) / wednesday (-1.73): Specific days of the week suggest timely, event-based reporting.

3. dont (-1.76): May appear in quotes or factual statements (e.g., "experts don’t agree"), common in balanced reporting.

Key Takeaways

- Fake news terms often involve politicized language, indirect sourcing, or speculative phrasing.

- Real news leans on attribution("said"), specific timelines(days of the week), and measured language.

- Caveats: The model may overfit to stylistic quirks (e.g., "via" in URLs) or dataset biases rather than inherent truthfulness. Always validate with context and additional fact-checking.

The results reveal a striking contrast between linguistic cues associated with fake and real news, underscoring how politicized language and indirect sourcing often signal low credibility. Words like "gop," "obama," and "sen" (likely referencing political figures or parties) dominate the fake news predictors, suggesting that polarizing or partisan narratives may correlate with misinformation. Terms such as "via" (implying indirect attribution) and "imag" (hinting at speculation or manipulated visuals) further highlight the lack of verifiable sourcing common in fake news. Conversely, real news indicators prioritize accountability and specificity: "said" reflects proper attribution, while references to days like "thursday" or "wednesday" align with timely, event-based reporting. The model’s reliance on these patterns, however, carries risks—terms like "dont" or "via" might reflect stylistic quirks (e.g., URLs or quotes) rather than inherent credibility. While these insights offer a useful heuristic, they also caution against overinterpreting isolated words without context, emphasizing the need to pair algorithmic analysis with critical human evaluation to discern intent and nuance.

This analysis highlights how linguistic patterns can signal credibility, though human judgment remains essential to avoid misinterpretation.

Conclusion

This pipeline achieves 98.9% accuracy using Logistic Regression with binary vectorization. The model effectively identifies political keywords and proper nouns as fake news markers, while neutral verbs and dates indicate real news. This project isn’t just about algorithms, it’s about restoring trust in information. By combining robust NLP techniques with interpretable models, we’ve built a system that flags fake news with very good results for such a basic approach. While challenges remain, the results prove AI can be a powerful ally in the fight against misinformation.