Predicting AI-Generated Misinformation in Social Media Posts

Thaddeus Felten (jdn7eh) - 3:30PM Section

Charles Stewart (gng4fn) - 2:00PM Section

STAT 4630

Introduction

In the digital age, there has long been the issue of misinformation online. Whether on social media, shady news sources, or illegitimate websites, the internet has made it incredibly easy to deceive and mislead. With the recent surge in generative artificial intelligence, this problem has been exacerbated greatly. It is easier than ever to spread misinformation online. Individuals can autonomously both create and upload posts including false information using generative AI. These posts can be used for any number of reasons: swaying public opinion, influencing health or consumer behavior, creating political or social discourse, etc. Because of this, we need ways to screen and identify AI-generated misinformation; it is both a technical challenge and social priority.

In this project, we employ machine learning techniques to classify social media posts as authentic, human-written content or AI-generated misinformation. The data is a mixture of content characteristics, behavioral signals, engagement metrics, and text attributes. The numerous variables reflect the complexity and multifaceted nature of misinformation detection. We seek to evaluate predictive performance for a set of machine learning models, ranging from an interpretable linear model, to tree-based methods, to large margin classifiers. Additionally, we analyze how the models vary in interpretability, flexibility, and features. By evaluating these models under one machine learning pipeline and evaluation framework, we aim to identify which models are best for this dataset and what sort of limitations there are in the data.

Dataset Description

The data that we used is a synthetic misinformation detection dataset that we sourced from Kaggle. It contains 500 social media posts that are represented with 31 variables of varying data types. Though the data is synthetic, it emulates common characteristics found in AI-generated misinformation and allows for controlled experimentation.

The predictors fall into a few broad categories: temporal features, text-based metadata, quantitative engagement metrics, content quality indicators, and fact-checking signals. Temporal features include time-stamp focused data like date, time of posting, month, and day of week. These variables can be used to determine if posting behavior and time-of-day patterns correlate with misinformation. Text-based meta data refers to categorical data such as platform, region, language, and topic. These variables allow the models to identify geographical and linguistic trends and account for differences across social media platforms. Quantitative engagement metrics refers to likes, shares, comments, click-through rate (CTR), and views. These variables display how a piece of content was interacted with and can indicate if misinformation spreads more rapidly than authentic information or if misinformation tends to have unique ratios of engagement metrics. Content quality indicators are linguistic and readability signals such as sentiment polarity, toxicity score, and readability index, which demonstrate a post's clarity, tone, and civility. Lastly, fact-checking signals include credibility source score, manual check flag, and claim verification status, which provides information on the post’s reliability. Our target variable, is_misinformation, is a binary label where 0 is authentic, and 1 is misinformation.

Although the data is synthetic, it is specifically designed for use in machine learning, deep learning, NLP, data visualization, and predictive analysis research. It provides a controlled environment to experiment with various machine learning methods, and includes a wide range of data types to work with. However, because it’s synthetic, the data may not capture some real world complexities in social media posts. Thus, while it’s suitable for comparing models and getting a general idea of predictive power, the performance metrics should not be interpreted as the true upper bounds for misinformation detection.

Exploratory Data Analysis

Through our initial exploration of the dataset, we could see that it is clean and well-structured. There were no missing values in any column, which made processing the data much simpler. For our variable of interest, is_misinformation, the classes were nearly evenly distributed, though there were a few more positives than negatives. The numeric variables had reasonable variability and did not have any major outliers. A correlation heat map indicated that the predictors are only weakly related to the response variable, indicating that models capable of capturing non-linear relationships may be more effective at prediction. The categorical data was well distributed across categories, which provided contextual diversity in the data. Overall, our EDA suggested that the data is high-dimensional, heterogenous, and weakly linearly separable. This indicates that tree-based methods or SVMs may be advantageous for predicting is_misinformation.

Preprocessing & Feature Engineering

Before feature engineering and model training, we performed a structured data-cleaning pass to verify dataset integrity (as mentioned above) and ensured all variables were usable in the downstream pipelines.

We verified data completeness using df.info() and df.isna().sum(), confirming that all 500 rows and 31 variables contained no missing values, so no imputation methods were needed. All columns also had appropriate data types for modeling: numeric features (e.g., toxicity_score, author_followers, engagement) were already in numeric form for scaling, categorical variables (platform, country, factcheck_verdict) were stored as object types for one-hot encoding, and the text field remained as raw strings for TF-IDF processing. Identifier and time-related fields (id, post_id, timestamp) were stored as strings and excluded from modeling since they carry no predictive signal. Because each variable was already typed correctly, no data-type corrections were required.

We removed non-predictive identifier fields (id, post_id, author_id) because they provide no modeling signal and would otherwise create hundreds of meaningless one-hot encoded categories. Time-related columns (timestamp, date, time, month, weekday) were also excluded, as they function primarily as metadata and do not offer interpretable features without additional transformation; deriving time-based variables could be explored in future work. The main text field required no manual cleaning, since TF-IDF inherently manages tokenization, lower-casing, and stop-word removal during preprocessing.

We prepared the categorical features using one-hot encoding, allowing each category to be represented as its own binary column without imposing any artificial ordering; this was implemented inside the project's ColumnTransformer alongside numeric scaling and TF-IDF text processing, with handle_unknown="ignore" to safely manage unseen categories. All numeric features were standardized with StandardScaler so that variables measured on larger scales (such as follower counts or engagement) would not dominate smaller-range features, and this step was especially important for models like logistic regression and SVM that rely on distance-based or gradient-based optimization. For the text field, we applied TF-IDF with a capped vocabulary of 1,000 tokens, using unigrams and bigrams and automatically removing English stop words, which helped highlight informative patterns while keeping the feature space manageable; placing TF-IDF within the ColumnTransformer ensured consistent preprocessing and prevented information leakage by fitting the vocabulary only on the training data. Although interaction features can capture nonlinear relationships, we chose not to include them because the combination of one-hot encoding and TF-IDF already produces a high-dimensional feature space relative to our sample size; adding interactions would increase overfitting risk, and nonlinear models such as boosting and random forests are already capable of learning interaction effects implicitly.

We used a 70/30 train–test split, giving us 350 training samples and 150 test samples, enough data for the models to learn while still holding out a clean set for evaluation. Because the dataset is slightly imbalanced, we applied stratified sampling to preserve the class proportions in both sets (roughly 53–54% misinformation). A fixed random seed ensured the split was reproducible. This simple, balanced setup provided a solid starting point before introducing cross-validation in the later modeling phases.

Models

To evaluate different approaches for detecting misinformation, we trained multiple models ranging from simple linear baselines to more flexible nonlinear methods and modern ensemble techniques. Each model was built using the same preprocessing pipeline (scaling numeric features, one-hot encoding categorical variables, and applying TF–IDF to the text field) to ensure a fair comparison. By keeping preprocessing and model training tied together in scikit-learn pipelines, we avoided data leakage and guaranteed consistent transformations across all cross-validation folds. The following sections describe each model, why it was chosen, and how it was implemented.

Logistic regression serves as our baseline model for predicting whether a post contains misinformation, and appeals for its simplicity and interpretability. This model used L2 (ridge) regularization by default, which helps stabilize coefficients in the high-dimensional feature space created by our text and categorical expansions. On the test set, logistic regression reached an accuracy of 0.527 and an ROC-AUC of 0.495, indicating limited performance as a linear model but providing a clear point for evaluating more flexible approaches in later models.

Next, we trained a decision tree classifier (using the same preprocessing pipeline as before). Decision trees can capture nonlinear relationships, but are also prone to high variance. To reduce overfitting, we limited the depth of the tree to 10 and required at least 5 samples per leaf. On the test set, the decision tree reached an accuracy of 0.487 and a ROC-AUC of 0.506, slightly below our logistic regression baseline. Precision and recall were balanced but overall low, suggesting the model had difficulty forming stable rules in the high-dimensional feature space. This result demonstrated why ensemble methods (like random forest or boosting) are preferred when working with sparse text features and nonlinear patterns.

Our third model was a random forest classifier, which improves on single decision trees by averaging many of them together. This technique aimed to reduce correlation among trees and lower overall variance. Our model used 300 trees with a small leaf size to limit overfitting while still allowing the ensemble to capture nonlinear patterns. On the test set, the random forest reached an accuracy of 0.540 and a ROC-AUC of 0.520, a small improvement over the standalone tree. The model tended to classify most posts as misinformation, leading to a high recall for that class but many false positives. We also looked at the model's feature importances to see which signals the forest relied on most. The plot in our code demonstrates how broad number features (e.g., text_length, toxicity_score) were the most influential, while individual TF-IDF terms contributed much less.

Our fourth model was an XGBoost classifier, aiming to better our tree methods by adaptively reweighting hard-to-classify examples. Our model used 300 boosting rounds with moderate depth and subsampling to control overfitting. On the test set, XGBoost achieved an accuracy of 0.547 and a ROC-AUC of 0.539, making it our strongest tree-based model so far, though only a modest improvement over random forest. Again, we explored top feature importances, and found that XGBoost relied heavily on certain city indicators and the TF–IDF phrase "ai misinformation," which suggests the synthetic dataset embeds strong associations between these features and the misinformation label. Numeric features like text length and toxicity also played meaningful roles, giving the model a bit more flexibility than our earlier tree methods.

Lastly, our fifth model was a linear SVM, functioning as another high-dimensional linear model. We set kernel=”linear” because linear SVMs tend to perform well with large feature spaces such as TF-IDF. On the test set, the linear SVM achieved an accuracy of 0.548 and a ROC-AUC of 0.511, with strong recall for the misinformation class but many false positives for non-misinformation posts. This pattern reflects how the margin-based objective prioritizes separating the positive class in a high-dimensional space. Overall, the SVM performed similarly to XGBoost in accuracy but showed limited improvement over the earlier models in our dataset.

Tuning & Cross-Validation

To refine model performance beyond the baseline results, we applied hyperparameter tuning using 5-fold stratified cross-validation. Stratification ensured that each fold preserved the original class balance, providing more stable AUC estimates for our slightly imbalanced dataset. All tuning took place within pipelines that included the full preprocessing stack (scaling, one-hot encoding, and TF-IDF) so that each fold replicated the complete training process and avoided data leakage.

We tuned three models chosen for complementary strengths: logistic regression as an interpretable linear baseline, linear SVM as a strong high-dimensional classifier well suited for text-heavy data, and XGBoost as a flexible nonlinear model capable of capturing complex interactions. Logistic regression was tuned by searching over the regularization strength C. GridSearchCV evaluated five values, and the best model selected $C = 0.01$ with a cross-validated ROC-AUC of 0.469. For linear SVM, we used an analogous grid over C, producing an optimal value of 0.1 and a best CV AUC of 0.476. Because boosting models are more sensitive to configuration, XGBoost was tuned using RandomizedSearchCV across 25 sampled hyperparameter combinations (learning rate, max depth, number of estimators, subsampling, and feature subsampling). The best configuration achieved a cross-validated ROC-AUC of 0.512, the strongest among the tuned models.

Evaluation and Interpretation

Model performance was evaluated using accuracy, precision, recall, F1-score, ROC–AUC, and PR–AUC to capture both classification quality and the models' ability to rank posts by misinformation likelihood. Because the target variable's distribution is fairly balanced, the additional metrics besides accuracy are essential to evaluating model performance. Across baseline models, performance was modest, which reflects the non-linearity of the data.

The logistic regression model performed reasonably as a baseline model, but had a ROC-AUC score less than 0.5. The standalone decision tree underperformed as well, having high variance and low accuracy. Random forests saw some improvement with an accuracy of 0.5400, but it had a recall of 1.000. This suggests that the random forests over-estimated the positives, coming at the cost of precision. The XGBoost model was more balanced, with better accuracy, precision, and F-1 scores than the simpler models, as well as ROC-AUC and PR-AUC. Tuning the XGBoost model achieved the highest ROC-AUC (0.558) and PR-AUC (0.591) among all models. This indicates it was best at ranking misinformation posts higher than authentic posts.

The SVM model had mixed results. Though the baseline model achieved a fairly high recall (0.7875), its ROC-AUC was low. Interestingly, tuning the SVM actually was detrimental, reducing all performance metrics across the board. The largest improvements seen from tuning were from the logistic regression model. Tuning the logistic regression model improved all performance metrics aside from PR-AUC. The tuned logistic regression model also achieved the highest accuracy among all models (0.567).

Results:

Model	Accuracy	Precision	Recall	F1 Score	ROC-AUC	PR-AUC
Logistic	0.526667	0.548387	0.6375	0.589595	0.4925	0.522097

Regression						
Decision Tree	0.486667	0.517647	0.55	0.533333	0.505804	0.531295
Random Forest	0.54	0.536913	1	0.69869	0.519643	0.550868
XGBoost	0.546667	0.565217	0.65	0.604651	0.539464	0.555005
Linear SVM	0.546667	0.552632	0.7875	0.649485	0.511429	0.542085
Logistic Regression (Tuned)	0.566667	0.568807	0.775	0.656085	0.501429	0.51511
Linear SVM (Tuned)	0.52	0.545455	0.6	0.571429	0.504643	0.514579
XGBoost (Tuned)	0.54	0.567901	0.575	0.571429	0.558214	0.591484

From our results, we can see that none of the models are particularly exceptional at predicting misinformation. This reflects the difficulty of the task and the datasets design. However, it is still clear that some models performed better than others. Tuned XGBoost performed the best on ranking based metrics, random forest achieved the highest recall, and the tuned logistic regression model had the highest accuracy. In the context of the dataset, we can determine that the predictors likely have non-linear relationships with the response variable. The variables may not have strong predictive power, or we may need to use more complex machine learning methods with more inputs to capture the relationships.

Discussion & Conclusion

Our results demonstrate that determining whether a post is misinformation or not is a difficult task, with our best models performing only slightly better than randomly guessing. Although the feature set is numerous and diverse, weak linear correlations and dispersed signals show that misinformation is not easily separable using simple decision boundaries. Ensemble methods like XGBoost and random forests performed slightly better than the others, however not by enough to be considered to have any real predictive power. Even the best performing models achieved moderate AUC scores, emphasizing the difficulty in predicting misinformation.

There are several directions we could go next. We could use more expressive text representations, to help capture semantic structure that TF-IDF is not able to. Alternatively, we could use additional feature engineering to try to amplify weak signals, particularly for interactions involving credibility, sentiment, and synthetic detection scores. Additionally, a larger dataset or a more complex synthetic generation process could help improve model training. Lastly, the natural next step is to implement neural networks which were not used in this project. A feedforward network or a transformer model may be better suited for identifying non-linear trends in high-dimensional data. While our current results highlight the difficulty in misinformation identification, they also create a foundational modeling pipeline to explore different approaches in future work.

Note on AI Usage: we used AI resources for debugging and refining our code.