Classifying Far-Right Extremism in Social Media: A Supervised Learning Approach

By Grace Kluender, Ethan Evans, and Evan Fantozzi

Introduction

As political polarization has grown in recent years, much attention has been given to the role of social media in disseminating and spreading extremist views[1]. We have learned that extremist messaging on these platforms can have dramatic consequences, including inciting hate crimes and other forms of targeted violence[2]. Online moderation, although important, has not served as a catch-all solution for this issue, as fringe platforms allow full-fledged extremist views to flourish while major platforms permit more subtle messaging.[3] Our project is centered within this context, as researchers and policymakers have pointed to machine learning-based solutions as one potential tool to identify and combat online extremism.[4]

Our model addresses a simple but important question - is a given text (in English) likely to be far-right extremist? By applying this tool to batches of text-based social media posts, public interest researchers or other concerned citizens could more quickly identify corners of the internet that are likely to cause real-life damage. The rest of this report outlines our data sources, model specifications, and our results, including surprising findings.

Data (n = 4670)

To train our model, we first obtained an existing dataset of 100,000 social media posts with text, scraped from explicitly Nazi and other alt-right online accounts from 2017[5]. We then randomized the list of posts and kept the first 1500 that we manually labeled as

[1] Shaw, A. (2023). Social media, extremism, and radicalization. *Science Advances, 9*(35), eadk2031. https://doi.org/10.1126/sciadv.adk2031

[2] Scrivens, R. (2024). Examining online indicators of extremism among violent and non-violent right-wing extremists. *Terrorism and Political Violence, 36*(9), 943–965. https://doi.org/10.1080/1057610X.2024.2347860

[3] Wang, M. (2025, May 15). *How extremist groups navigate the online ecosystem: A Q&A with SIPA Professor Tamar Mitts*. Columbia SIPA. https://www.sipa.columbia.edu/news/how-extremist-groups-navigate-online-ecosystem-qa-sipa-professor-tamar-mitts

[4] University of Technology Sydney. (2023, February 27). *Can algorithms catch online extremism before it takes hold?* UTS Newsroom. https://www.uts.edu.au/news/2023/02/can-algorithms-catch-online-extremism-it-takes-hold

[5] Islet, S. (2017, November 14). *Nazi Tweets* [Data set]. Kaggle. https://www.kaggle.com/datasets/saraislet/nazi-tweets

extremist, based on methodology from the alt-right Twitter Census[6]. We sought to supplement these posts with a newer dataset (and corresponding lexicon), but were unable to locate publicly available aggregated datasets consisting of posts after the January 6th invasion, an event that fundamentally changed how far-right messaging has operated in the United States and across the world.[7] To address this gap, we identified 51 additional posts from recent years that fulfilled the alt-right Twitter Census criteria using TwitterAPI.io, an enterprise web scraping tool[8]. We would have liked to include a larger number of recent extremist posts, but we were limited by the time-intensive nature of the task and API credit limits.

To balance our model, we obtained a list of 1.6 million random tweets from 2009[9], and similarly randomized and kept the first 1500 that we manually labeled as not extremist. We supplemented these data with an additional random 1619 posts from 2024 and 2025 that addressed topics such as sports, politics, and Donald Trump, all manually labeled as non-extremist (though many expressed right-wing views)[10]. We included these non-extremist tweets–particularly those relating to politics and Donald Trump–to diversify our dataset and enable the model to better differentiate between mainstream right-wing political content and far-right extremist rhetoric. Again, these supplemental posts were obtained using TwitterAPI.io.

All posts were cleaned similarly, removing URLs, emojis, extra whitespace, and punctuation beyond question marks, exclamation points, commas, and periods. This included removing hashtags, though the actual text included in the hashtags was not removed. Posts that were empty or not in English were discarded.

Feature Engineering

In our feature engineering pipeline for detecting far-right extremist content in tweets, we use a combination of linguistic, stylistic, and semantic features, which are designed to capture a unique aspect of the language used in extremist social media posts. Below we describe each feature, how it was engineered, and its relevance to predicting far-right extremism.

[6] Berger, J. M. (2018). *The alt-right Twitter census: Defining and describing the audience for alt-right content on Twitter*. VOX-Pol Network of Excellence. https://www.voxpol.eu/download/vox-pol_publication/AltRightTwitterCensus.pdf

[7] Atlantic Council. (2022). *After the insurrection: How domestic extremists adapted and evolved after the January 6 US Capitol attack*. Atlantic Council. https://www.atlanticcouncil.org/in-depth-research-reports/report/after-the-insurrection-how-domestic-extremists-adapted-and-evolved-after-the-january-6-us-capitol-attack/

[8] TwitterAPI.io. (n.d.). *Enterprise-grade Twitter data API*. https://twitterapi.io/twitterapi.io+9

[9] Kazanova. (2017). *Sentiment140 dataset with 1.6 million tweets* [Data set]. Kaggle. https://www.kaggle.com/datasets/kazanova/sentiment140

[10] TwitterAPI.io. (n.d.). *Enterprise-grade Twitter data API*. https://twitterapi.io/twitterapi.io+9

Our **Toxicity** feature is computed using a pre-trained *Detoxify* model[11], which assigns a score between 0 and 1 to indicate how rude, disrespectful, or aggressive a post is. This is particularly useful for detecting hostile language commonly associated with extremist content.

Our **Subjectivity** feature is extracted using *TextBlob*[12], which evaluates the degree to which a tweet reflects opinion or emotion, as opposed to factual reporting. High subjectivity scores may correlate with extremist-style rhetoric or emotionally charged messaging.

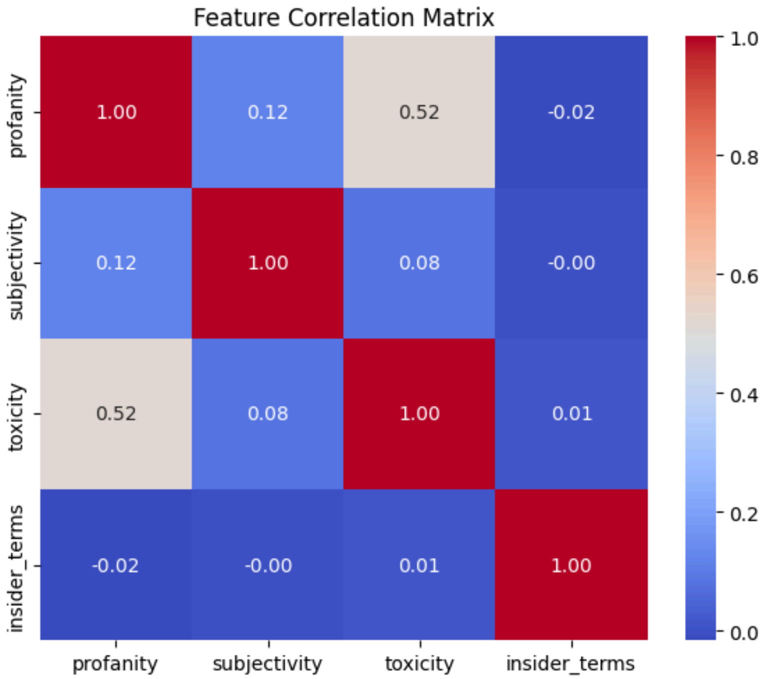
Next, we have our **Profanity** feature, which employs the *better_profanity*[13] library to return a binary flag indicating the presence or absence of profanity. To improve efficiency, our team stores the library's list of profane words in an external text file. This setup avoids reloading or reconstructing the list each time the feature is used, reducing overhead during processing.

We also included an **Insider Terms** keywords feature to flag highly-relevant extremist-related words compiled by our team.

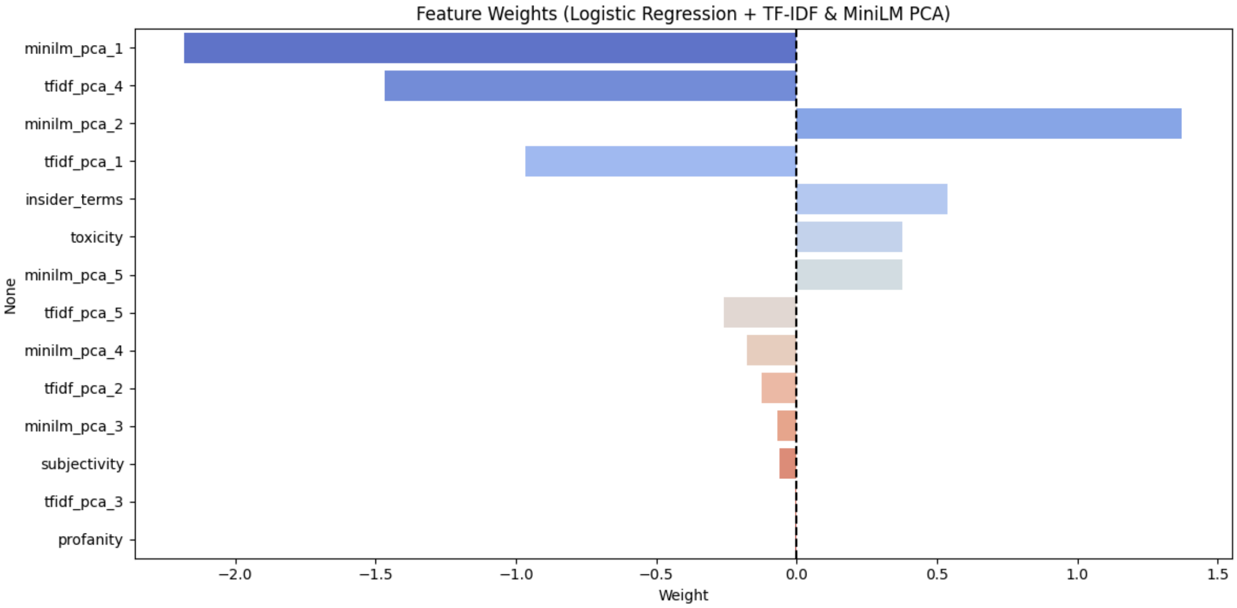
Finally, since classifying far-right extremism often requires semantic understanding beyond surface-level text, we incorporate two vectorization features to capture meaning and context. The first is **TF-IDF (Term Frequency–Inverse Document Frequency)[14]**, which assigns weights to unigrams and bigrams based on how frequently they appear in a tweet relative to their frequency across the entire dataset. The second is **MiniLM embeddings**, which are dense vector representations generated by a pre-trained SentenceTransformer model[15]. These embeddings encode the semantic content of each tweet in a high-dimensional space, allowing the model to detect meaning even when the exact words vary. These features allow for our model to identify more subtle expressions of extremism.

Initial testing showed no significant collinearity between our non-vectorized features, which helped us in deciding which features to keep.

[11] Hanu, D. (2020). Detoxify: Transformer-based toxicity scoring. PyPI. https://pypi.org/project/detoxify/
[12] Loria, S. (n.d.). TextBlob: Simplified text processing. TextBlob Documentation. https://textblob.readthedocs.io/en/dev/quickstart.html
[13] Nguyen, T. (n.d.). better_profanity: Fast, simple profanity filter for Python. PyPI. https://pypi.org/project/better-profanity/
[14] scikit-learn developers. (n.d.). *TfidfVectorizer*. https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
[15] Reimers, N., & Gurevych, I. (n.d.). *sentence-transformers*. https://www.sbert.net/



We also ran an initial logistic regression to get an initial sense of feature relevance via the feature weights. In order to directly compare *all* features, we ran a Principal Component Analysis on the vector-embedded features, simplifying them for the sake of analysis while still preserving as much information as possible. The results of the feature testing were as follows:



The top relevant features are MiniLM, TF-IDF, Insider Terms, and Toxicity, with Subjectivity and Profanity having less of an overall impact on the model.

Linear Model: Logistic Regression Model

For our actual logistic regression model (non-testing, without vector reduction), we applied L2 regularization to reduce overfitting by evenly shrinking coefficients while retaining all features, since extremist signals can be subtle and distributed. Through cross-validation, we tested several values of the regularization strength C and found that C=1.0 achieved the best performance, with 93% accuracy on the testing data. However, the recall for the extremist class, which reflects how well the model identifies all true extremist texts, was 88%. Because we prioritize this metric to avoid missing extremist content, this lower recall highlights the linear model's limitation in fully capturing extremist instances. This is why we ultimately decided to employ a nonlinear model for the final classifier, which we will discuss next.

Non-Linear Model: Support Vector Machine with RBF Kernel

We decided to employ a Support Vector Machine (SVM) with a nonlinear kernel due to its strong performance on high-dimensional feature spaces like those produced by our text analysis pipeline. We take advantage of the flexibility that SVMs offer through their tunable hyperparameters that tailor the model to the characteristics of our dataset.

Kernel Selection Through Cross-Validation

To determine the best nonlinear kernel, we evaluated three common kernels—polynomial, radial basis function (RBF), and sigmoid—using 5-fold cross-validation on the training set. We measured recall (our primary metric due to the importance of detecting extremist content), accuracy, and F1 score for each kernel. The RBF kernel consistently outperformed the others, achieving the highest recall and accuracy. Based on these results, we selected the RBF kernel for our final model.

The RBF kernel enables the SVM to learn nonlinear decision boundaries by implicitly mapping input features into a higher-dimensional space where classes can be separated linearly. It measures similarity between two points using:

$$K(x, y) = \exp(-\gamma ||x - y||^2)$$

Here, the gamma term controls how far the influence of a single training example reaches.

Hyperparameter Tuning and Model Optimization

To maximize recall while controlling for overfitting, we performed a grid search that optimizes recall over the following key hyperparameters:

- **C**, the regularization parameter, which determines how much we penalize misclassifications on the training data.
- **γ**, the parameter which controls the influence of individual training points.
- **class_weight**, the parameter that adjusts the penalty for misclassifying underrepresented classes.
- **shrinking**, which is either set to enable or disable the shrinking heuristic, an optimization technique used to speed up convergence by temporarily removing variables that are unlikely to become support vectors based on current estimates.

We found that the best parameter combination found was:

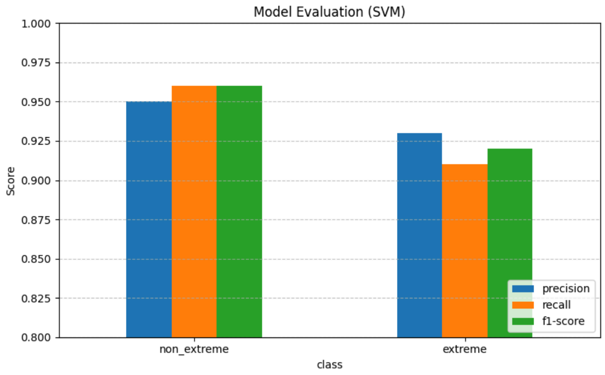
C = 10, class_weight = balanced, gamma = 0.1, shrinking = True

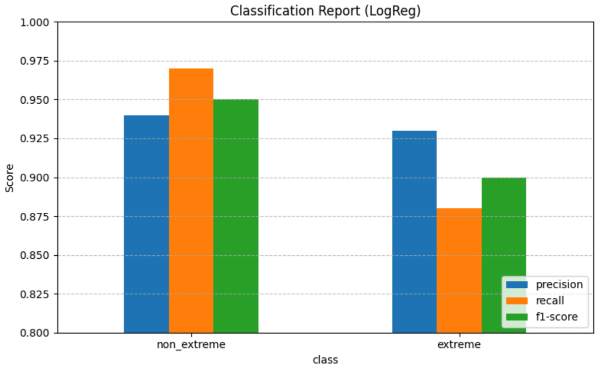
This model achieved a weighted recall score of approximately 0.90 during cross-validation.

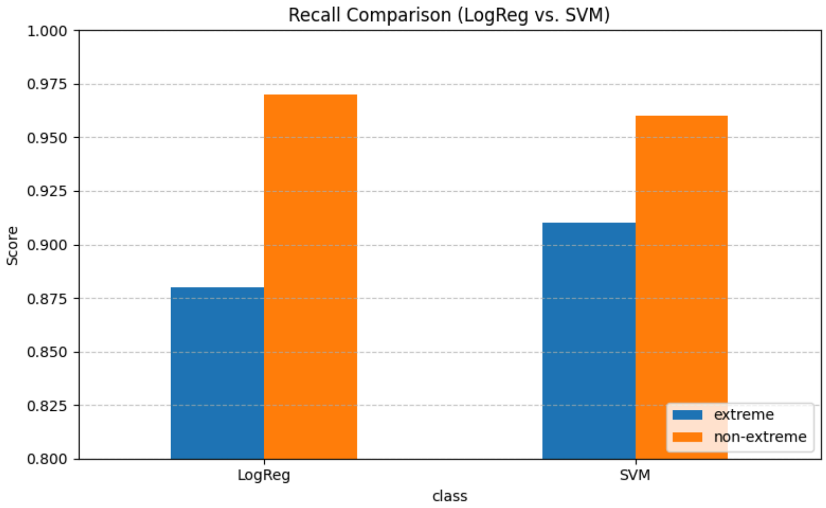
*It should be noted that the kernel selection process and hyperparameter tuning was conducted before integrating TF-IDF vectorization and MiniLM embeddings.

Results

Overall, both the Support Vector Machine (SVM) and Logistic Regression models performed well on both the training and testing datasets. SVM achieved a slightly higher accuracy of 95%, compared to 93% for Logistic Regression. Notably, SVM also outperformed Logistic Regression in terms of recall, with a score of 91% versus 88%. Given our emphasis on recall—prioritizing the identification of extremist content over minimizing false positives—this distinction is especially important. A detailed summary of the model performance metrics is provided below.







Additionally, when given brand-new user input, the model was able to predict both clear-cut and subtle far right rhetoric. In cases of common far-right terms being used in non-far right contexts, the model avoided becoming confused. Some contexts where the model did become confused by adversarial user input was when newer (post-2017) insider terms were used, when the model was given short input (less than three words), and when the input included highly salient far-right terms without surrounding context.

Surprises

One of the primary surprises was how critical context is in identifying far-right extremism. This is likely due to the fact that many terms and concepts associated with extremist rhetoric also appear in non-extremist discussions, so an extra layer of nuance is necessary for the model to make the correct classifications. Another surprise was the importance of timeliness. Far right language - including dog whistles and coded vernacular- are in constant flux. As a result, a model trained only on data from 2017 may struggle to recognize language patterns that are newly characteristic of far-right content in 2025.

Conclusion

This is the beginning of a useful classifier. In its current form, it is able to consistently detect far-right extremism, both on our testing data and on non-adversarial user input. However, this tool is not without limitations. To continue developing this project, we could:

1) Include a temporal feature that captures when a piece of content was written
2) Build out an even more robust dataset of recent and non-U.S. specific text samples (both extreme and normal), to better handle adversarial input
3) Update the Insider Terms list to include newer far-right dog whistles
4) Make this tool publicly available via an API and/or web application
5) Introduce account-scraping functionality for content moderation on newer social media platforms like BlueSky.