Machine Learning Final Project Report

Temuulen Byambabaatar
Email: tbyambabaatar@fordham.edu

Isaia Caluza
Email: icaluza@fordham.edu

Donald Ye
Email: dye10@fordham.edu

Testing Text Classification Models on Political Bias in Online Media and Investigating Potential Patterns in Political Bias and Misinformation

1. Problem Statement

Our project focuses on identifying political bias in online text content, particularly in news articles and social media posts. Understanding the ideological framing of content is essential, as it can subtly influence how the public perceives and interprets information. Detecting political bias in language can empower readers, researchers, and content moderators to assess information and its potential slant critically.

We define bias as an ideological leaning, such as left, center, and right. This definition prompts us to set a primary goal of classifying a given text based on its political orientation using supervised machine learning techniques.
We use supervised machine learning models such as Logistic Regression, Deep Neural Networks (DNN), and fine-tuned transformer models (DistilBERT) for political bias classification on labeled datasets. These models were trained on labeled datasets of political texts and even evaluated using accuracy, recall, F1 score, and ROC-AUC.

For feature extraction and text representation, we implemented TF-IDF vectorization and SimCSE. We first utilized TF-IDF to vectorize the words of news articles and online content with numerical scores. Moreover, SimCSE improves sentence-level understanding and analyzes the similarity between content and known ideological positions.

As an exploratory extension to our core classification task, we also applied unsupervised clustering (i.e., k-means and HDBSCAN) to visualize how political texts group together in embedding space based on their ideological leanings. Additionally, we briefly explore how a misinformation-labeled dataset could be integrated into future work to investigate potential correlations between political bias and misinformation. However, this remains outside the scope of our main predictive task and is proposed as a possible direction for future development.

By focusing on supervised political bias prediction while setting the foundation for

broader investigations into misinformation, our project contributes to detecting political bias in online discourse.

2. Datasets

Our primary dataset "siddharthmb/political-bias-prediction-media-splits" builds upon political bias ratings curated by AllSides.com. AllSides is a well-known platform that classifies media sources based on political bias using a multi-method approach that includes:

- Editorial Review
- Blind Bias Surveys of American readers
- Third-party data integration
- Independent expert analysis
- Community ratings and feedback

The dataset used in our project contains 30,246 news articles, each labeled with political ideology as left (0), center (1), or right (2). The labels are based on AllSides' aggregated and verified rating system. Each data sample includes the following fields:

- bias: The numerical label (0 = left, 1 = center, 2 = right)
- bias text: Human-readable bias description
- content: A cleaned and preprocessed version of the article text (excluding names and footers)
- title, url, source, date, authors: Additional metadata
- source_url: The original news website

The dataset is split into training, validation, and test subsets by media outlets. It means that articles from the same source do not appear in multiple subsets. This design choice is essential to prevent overfitting to the writing style or vocabulary of individual news outlets. It, also, helps test how well the model generalizes to previously unseen media/data which mirror real-world deployment scenarios.

The second dataset we used was the "roupenminassian/twitter-misinformation" dataset from HuggingFace. The dataset compiles several existing datasets to train misinformation detection models. Each instance in the dataset contains: (1) text: a string feature containing the tweet or news content, and (2) label: a binary classification (0 for factual, 2 for misinformation). The data is split into 92,394 training examples (60,309 factual and 32,805 misleading) and 10,267 testing examples (6,773 factual and 3,494 misleading).

3. Methodology

Model Selection: We experimented with a diverse set of models, each chosen to address different aspects of the classification task:

TF-IDF vectorization, Logistic Regression, and Deep Neural Networks (DNN)

This project's first approach was to preprocess (tokenize) article texts with the traditional vectorization method, TF-IDF. We, then, fed these TF-IDF vectors as inputs to our models.

Our first modeling attempt involved a simple, interpretable baseline: Logistic Regression. The Logistic Regression algorithm predicted political bias by applying a weighted sum over features, passing the result through a sigmoid function to compute class probabilities. This model was computationally fast and helped us inspect which terms/words (from the TF-IDF vector) most influenced predictions.

We recognized that linear classifiers are limited to linearity and cannot tackle nonlinear and complex meanings, which is essential in understanding political language. So, we utilized Tensorflow to develop a Deep Neural Network to model nonlinear interactions among TF-IDF features/words. The model's architecture included two hidden layers with "ReLU" activations, dropout regularization,"Categorical Cross-Entropy" loss function, and Adam optimizer for adaptive learning rate tuning.

Transformer Models: DistilBERT

We employed DistilBERT, a distilled version of BERT (Bidirectional Encoder Representations from Transformers). BERT is a deep learning model developed by Google that uses a bidirectional transformer architecture to simultaneously capture context from both the left and right sides of a token. It generates deep contextualized word embedding, which enables it to achieve state-of-the-art results on a wide range of natural language understanding tasks.

DistilBERT is this model except on a smaller scale; it retains approximately 97% of its performance while reducing model size by 40% and increasing inference speed by 60%. DistilBERT was fine-tuned on our labeled dataset for a three-way text classification task (left, center, right). Before model training, the textual data was preprocessed through tokenization, segmenting each input into subword units compatible with DistilBERT's tokenizer. The tokens were subsequently mapped to their corresponding numerical indices in DistilBERT's vocabulary through encoding. The encoded sequences were then passed through DistilBERT's transformer layers, which model complex contextual dependencies within the text. The final hidden states were fed into a classification head to output probabilistic predictions over the three target classes. Model performance was evaluated using standard metrics, including F1 score, recall, accuracy, and AUC-ROC.

hDBSCAN Clustering:

We tested clustering techniques to explore ideological groupings in an unsupervised manner. Given that we don't know the number of clusters and predict clusters to be arbitrary in shape, we implemented hierarchical Density-Based Spatial Clustering of Applications with Noise (hDBSCAN) to hopefully improve clustering results by visualizing based on its density and identifying outliers as noise.

SimCSE (Simple Contrastive Sentence Embedding)

In an effort to hopefully improve model performance in terms of text classification, Simple Contrastive Sentence Embedding (SimCSE) was implemented to fine-tune the vectorization of text. SimCSE is a powerful transformer model whose goal of contrastive learning is to cluster similar data points and 'push' away dissimilar ones in the embedding space. The input is a set of paired examples, where two data points in the pair are semantically related. The model outputs a vector representation with dimensions ranging from 768-1024, which are used to represent the semantic meaning of the sentence, yielding the ability to cluster sentences based on semantic similarity. It is not only concerned with the frequency of words, but how the words relate to each other in a sentence, making it better than TF-IDF. The process is twofold: (1) tokenization, and (2) embedding. The process starts with breaking down a sentence into individual words or sub-words called tokens. Each token is then converted into a numerical representation called a word embedding, which captures the semantic meaning of the word. The type of embedding technique varies but transformer-based models like BERT employ "self-attention mechanisms" which aim to understand the relationships between words in a sentence. Thus, similar sentences are represented by vectors that are close to each other in the high-dimensional space, making it easy to compare and cluster sentences based on their meaning.

4. Results

Logistic Regression

Table 0 - Dataset bias class distribution

	left	center	right
Train	8861	7488	10241
Valid	1640	618	98
Test	599	402	299

Fig 1.1 - Top words of each political bias

Fig 1.2 Classification report of Validation Data and Test Data + ROC-AUC score

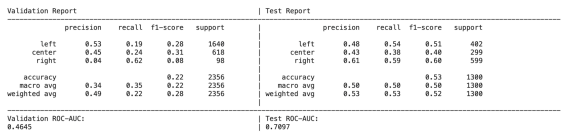
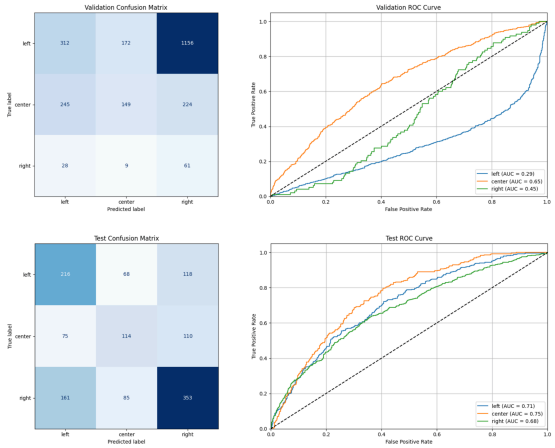


Table 0 highlights a significant class imbalance across training, validation, and testing datasets, with right-leaning articles heavily outnumbering center and left-leaning ones. This imbalance skewed model training toward the majority classes. *Fig 1.1* shows that the logistic regression model relied on source names (e.g., CNN, Reuters) and politically charged topics (e.g., illegal immigration, privacy rights) as strong political bias indicators.

The model achieved a low accuracy of 22% on validation data, suggesting limited predictive strength. The model showed notable behavior in terms of precision and recall. It showcased high recall (0.62) but extremely low precision (0.04), meaning it aggressively labeled articles as "right" with many false positives. For left-leaning articles, the model demonstrated higher precision (0.53) but lower recall (0.19). This suggests that the model was cautious in predicting "left," being relatively accurate, but missing many true left-leaning articles.

Testing results differed as accuracy rose to 53%, with precision (0.61) and recall (0.59) for right-leaning articles improving substantially. The model benefited from the imbalance since right-leaning articles were five times more common in the testing set than in validation. IN contrast, left-leaning articles were eighty times less represented. Overall, the results suggest the model performed well prospectively due to class imbalance, rather than generalizations.

Fig.2 - Confusion matrix + ROC curve



The confusion matrix suggests heavy confusion between the left and right classes. This indicates that there must be frequent overlap of vocabulary between the classes/political biases. Moreover, the validation data's macro-averaged ROC-AUC score (0.4645), below 0.5, suggests that TF-IDF + Logistic Regression alone cannot capture nuanced political bias in complex natural language. ROC-AUC curve analysis further reinforced these findings. The model does best at recognizing center-leaning articles (AUC = 0.65).

On the other hand, the model performs badly when predicting right (AUC = 0.45) and left (AUC = 0.29). However, as mentioned before, the testing data outperforms the validation data due to class imbalance. Hence, we made the rest of our evaluation on validation data.

Deep Neural Networks

Fig.3 - Classification report of Deep Neural Networks

	precision	recall	f1-score	support
left	0.57	0.23	0.33	1640
center	0.35	0.32	0.34	618
right	0.05	0.52	0.08	98
accuracy			0.27	2356
macro avg	0.32	0.36	0.25	2356
weighted avg	0.49	0.27	0.32	2356

Macro-Averaged ROC-AUC Score: 0.4751

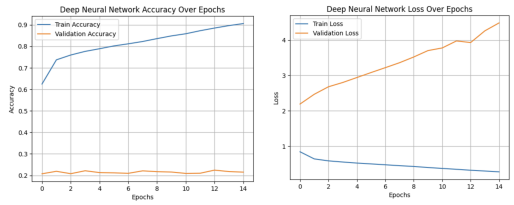
The Deep Neural Network was trained to better model political language's nonlinear and complex relationships.

After training the DNN, the classification report demonstrates that there was a modest improvement in validation accuracy, reaching approximately 27% (in comparison to Logistic Regression's 22%). Also, a slight improvement in ROC-AUC with 0.4751 compared to Logistic Regression 0.4645. Despite these improvements, critical findings emerge from the epoch reports.

Fig. 4 - Training accuracy and loss, validation accuracy and loss over 15 epochs

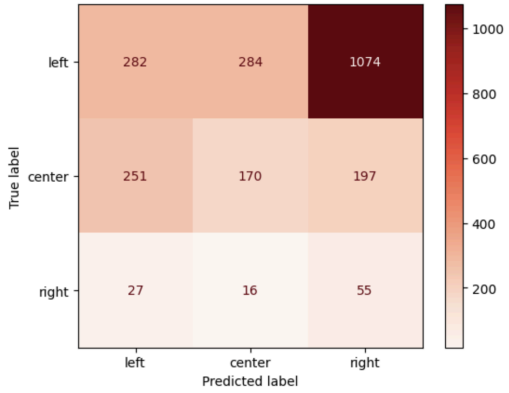
```
Epoch 1/15
416/416 ━━━━━━━━━━━━━━━━━━━━ 4s 6ms/step - accuracy: 0.5384 - loss: 0.9557 - val_accuracy: 0.2076 - val_loss: 2.1917
Epoch 2/15
416/416 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.7365 - loss: 0.6502 - val_accuracy: 0.2190 - val_loss: 2.4675
Epoch 3/15
416/416 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.7690 - loss: 0.5743 - val_accuracy: 0.2080 - val_loss: 2.6801
Epoch 4/15
416/416 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.7822 - loss: 0.5418 - val_accuracy: 0.2216 - val_loss: 2.8003
Epoch 5/15
416/416 ━━━━━━━━━━━━━━━━━━━━ 2s 4ms/step - accuracy: 0.7991 - loss: 0.5075 - val_accuracy: 0.2131 - val_loss: 2.9413
Epoch 6/15
416/416 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.8045 - loss: 0.5001 - val_accuracy: 0.2118 - val_loss: 3.0824
Epoch 7/15
416/416 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8166 - loss: 0.4669 - val_accuracy: 0.2097 - val_loss: 3.2202
Epoch 8/15
416/416 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8238 - loss: 0.4434 - val_accuracy: 0.2211 - val_loss: 3.3600
Epoch 9/15
416/416 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8389 - loss: 0.4223 - val_accuracy: 0.2173 - val_loss: 3.5221
Epoch 10/15
416/416 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8589 - loss: 0.3853 - val_accuracy: 0.2156 - val_loss: 3.7029
Epoch 11/15
416/416 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8647 - loss: 0.3618 - val_accuracy: 0.2093 - val_loss: 3.7770
Epoch 12/15
416/416 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8780 - loss: 0.3376 - val_accuracy: 0.2101 - val_loss: 3.9751
Epoch 13/15
416/416 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8905 - loss: 0.3108 - val_accuracy: 0.2245 - val_loss: 3.9299
Epoch 14/15
416/416 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.8998 - loss: 0.2920 - val_accuracy: 0.2177 - val_loss: 4.2609
Epoch 15/15
416/416 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.9121 - loss: 0.2678 - val_accuracy: 0.2152 - val_loss: 4.4852
```

Fig. 5 - Graph visualization of accuracy and loss over 15 epochs



The epoch report and the curves represent the model's overfitting. The training accuracy improved significantly with each epoch, but the validation accuracy lagged, and the validation loss increased. This indicated that the DNN was overfitting the training data, memorizing patterns without generalizing well to new, unseen examples. While the DNN performed slightly better than logistic regression, the magnitude of improvement was small.

Fig. 6 - Confusion matrix of Deep Neural Networks



The confusion between left and right biases persisted. Although the DNN could model more complex interactions between words, it is still fundamentally constrained by the limitations of TF-IDF features. Conclusively, without richer input presentations, the model will struggle to fully disambiguate closely related political language. These findings suggest that while DNNs are theoretically more powerful than linear models, their practical benefits are limited if the input features are shallow or context-poor. Substantial improvements would require richer, more context-aware embeddings like those produced by Transformer models.

DistilBERT Results

Fig. 7.1 - Classification report of DistilBERT Test-Data

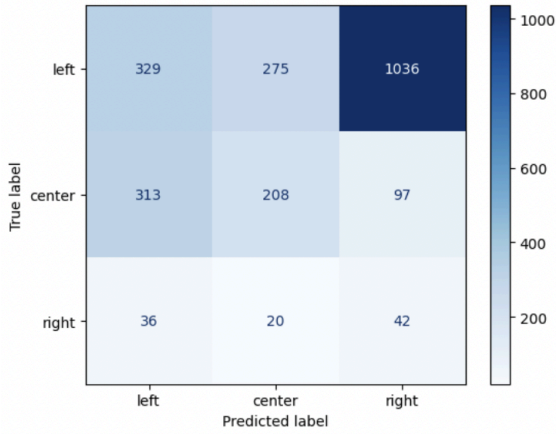
	precision	recall	f1-score	support
left	0.39	0.63	0.48	402
center	0.17	0.14	0.15	299
right	0.62	0.41	0.49	599
accuracy			0.42	1300
macro avg	0.39	0.39	0.37	1300
weighted avg	0.44	0.42	0.41	1300

Fig. 7.2 - Classification report of DistilBERT Validation-Data

	precision	recall	f1-score	support
left	0.48	0.19	0.28	1640
center	0.38	0.32	0.35	618
right	0.03	0.42	0.06	98
accuracy			0.24	2356
macro avg	0.30	0.31	0.23	2356
weighted avg	0.44	0.24	0.29	2356

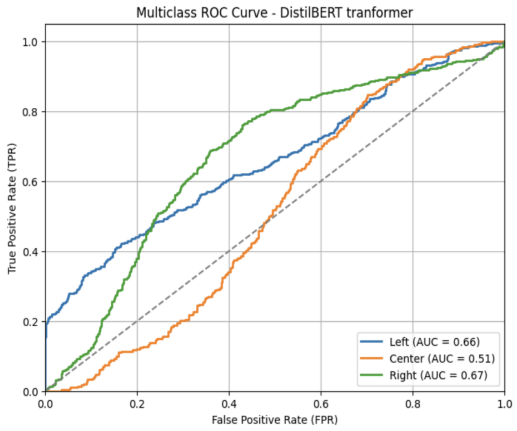
The DistilBERT transformer model was trained to be better than the DNN and Logistic Regression models above. By leveraging its pretrained representations, DistilBERT could model the language more effectively than traditional machine learning approaches. The classification report reveals an improvement in validation accuracy of approximately 42%. It is also essential to consider that the test data is imbalanced, emphasizing the "Right" class. When examining *Fig.7.1* and *Fig.7.2,* the support size and the disparity between F1 scores in the test and validation data should be noted.

Fig.8 - Confusion matrix of DistilBERT



The confusion matrix above demonstrates a slightly better true positive rate for "left", "middle", and "right" compared to TF-IDF + Logistic Regression and DNN. Overall, they all suffer from an imbalanced validation set which explains the underperforming results. However, there's a gradual improvement through the progression of the three models.

Fig. 9 - AUC-ROC score of DistilBERT



The AUC for the "Left" class was 0.66, indicating a moderate ability to distinguish it from other classes, while the "Center" class was 0.51, suggesting poor classification performance. The "Right" class achieved an AUC of 0.67, which reflected a reasonable ability to distinguish it. These results showed that the model performs well in the left and right classes but struggles in the center class.

Clustering Results

Fig. 10 - 2D & 3D HDBSCAN + SimCSE clustering visualization of political bias data.

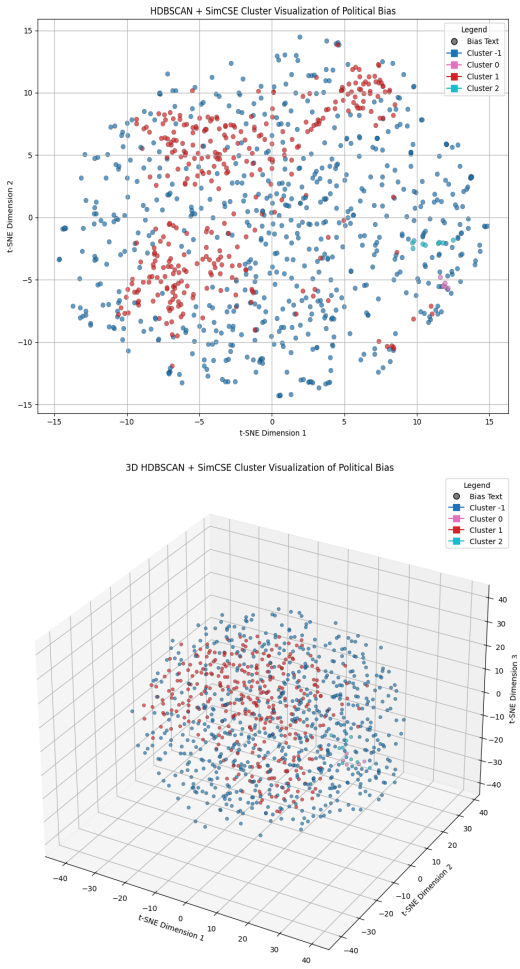


Table 1 - Hierarchical DBSCAN clustering summary on SimCSE embeddings of text from the political bias dataset.

Cluster	True Bias		
	Left	Center	Right
-1	221	193	235
0	1	0	4
1	112	103	122
2	2	2	5

Fig. 11 - 2D & 3D HDBSCAN + SimCSE clustering visualization of misinformation data.

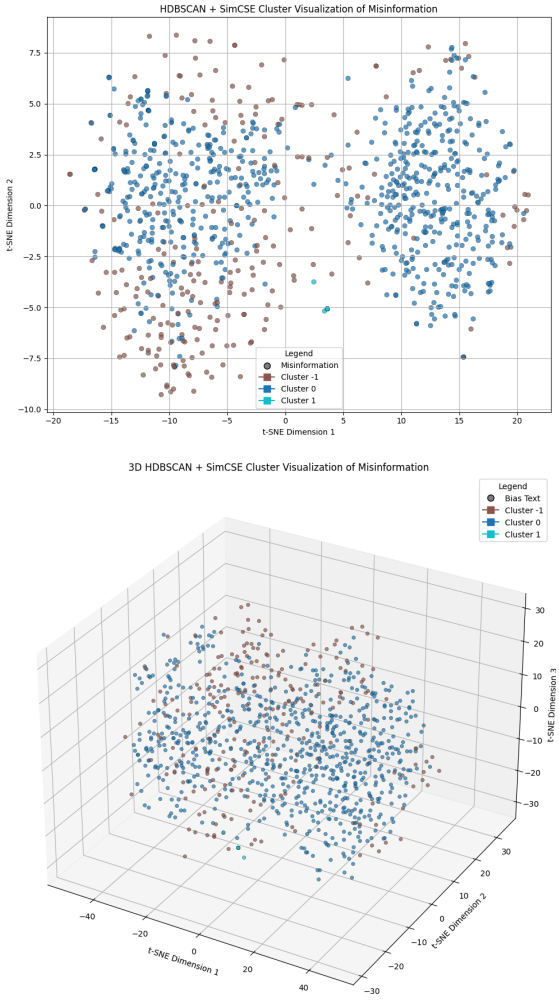


Table 2 - Hierarchical DBSCAN clustering summary on SimCSE embeddings of text from the misinformation dataset.

Cluster	True Label	
	Factual (0)	Misinformation (1)
-1	179	64
0	499	248
1	0	10

The most significant clusters for the political bias dataset are Cluster -1 and Cluster 1. Cluster -1 contains 221 left, 193 center, and 235 right-biased texts (see Fig. 10 & Table 1). Cluster 1 contains 112 left, 103 center, and 122 right. Both clusters contain a somewhat even distribution of left, right, and center points, showing no significant pattern in the clusters. Clusters 0 and 2 are categorized as noise, as very few data points are within the clusters. As a result, the DBSCAN + SimCSE approach was ineffective in capturing significant text patterns among the three bias labels, meaning no specific SimCSE embedding of a text is particular to a bias. Thus, we cannot confidently say that text meanings differ across the three political groups. The clustering algorithm identified three clusters for the misinformation dataset: Cluster -1 contains 179 factual and 64 misinformation data points, suggesting samples may be less semantically coherent. Cluster 0 is the largest and most dense cluster, containing 499 factual and 248 misinformation data points; the 2:1 ratio may reflect the dataset's class imbalance. Lastly, Cluster 1 is the smallest cluster, containing only misinformation, suggesting that a localized embedding region with similar semantic content may have a pattern of misinformation.

Fig. 12 - Wordclouds by cluster show the top 50 words in each cluster with TF-IDF for political bias data.

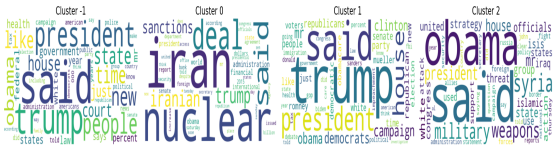
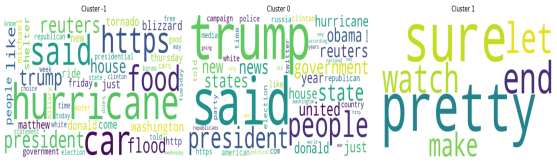


Fig. 13 - Wordclouds by cluster show the top 50 words in each cluster with TF-IDF for misinformation data.



To evaluate poor clustering performance, Wordclouds were generated and implemented using TF-IDF to identify the top words within each cluster. Shown in *Fig. 12*, almost all clusters in the political bias dataset contain the word 'said' as a top word, which is an insignificant word in terms of contributing to the political bias of a sentence. Thus, it may explain why the clustering was not as effective because it took into account non-significant filler words. For the misinformation data, Fig. 13 shows more meaningful words per cluster like 'Trump' in Cluster 0, but still captured most semantically meaningless words like 'said' again, demonstrating that TF-IDF is not the best method for extracting relevant words. Future development of this project would involve greater awareness of top words and removing words that have no significant semantic contribution to the sentence.

5. Discussion

Challenges

Table 3 - Dataset Splits in Percentage

Set	Left	Center	Right
Training	33%	28%	39%
Validation	70%	25%	5%
Test	30%	23%	46%

We encountered several challenges throughout this project. First, the class imbalance in our datasets led to biased learning and evaluation. Right-leaning articles dominated the training and testing sets, skewing the model's predictions toward the majority class. This imbalance (*Table 3*) made it difficult for the model to recognize underrepresented labels such as leftist or centrist content. Second, the limitations of TF-IDF became apparent during modeling. TF-IDF could not capture word order, synonymy, or contextual meaning, significantly hindering the models' ability to learn nuanced political rhetoric. And as verified by the word clouds of the TF-IDF vectors, many similar top words across the clusters were filler words. While it served as a valid baseline representation, it lacked the depth required for capturing ideological tone or rhetorical framing.

Limitations

Several limitations shaped our outcomes. The project faced limitations in both data and model design. The datasets lacked coverage of specific topics (e.g., foreign policy, minority activism). This restricted our ability to test generalization across diverse issue areas, and even geographic locations. Furthermore, while transformer models working on raw text offered slightly better performance, they still struggled to predict political bias successfully. Our task demanded subtle distinctions that even large pre-trained models found difficult without extensive fine-tuning and contextual understanding. Finally, although we used SimCSE, a

state-of-the-art contrastive learning method, to embed sentence meaning, the model was not explicitly trained to disentangle political ideology from truthfulness. Thus, semantically similar statements could cluster together despite having opposite political leanings or factual reliability. This challenge was further exacerbated by noise from overlapping vocabularies in political speech.

Use of dimensionality reduction

We found that the implementation of t-SNE and HDBSCAN helps project high-dimensional embeddings like SimCSE into lower dimensions for visualization, which is helpful. However, even though we used all these reduction tools, the clustering plots showed natural clusters of the data and no distinct clusters among the bias or misinformation labels. As mentioned earlier, there was a lot of overlap of embeddings belonging to left, right, and center within the most significant clusters, so we could not say there were particular texts specific to any label.

Future Work

Future iterations of this project should prioritize a mix of normative and empirical content. The potential issue with attempting to find a correlation between political bias and misinformation via semantic clustering is that we must assume that politically biased texts contain an empirical claim, something we can verify. All misinformation texts contain objective information that can be verified as factual or misleading. However, we did not check if the politically biased texts all contain a verifiable claim; they potentially mostly have normative claims/opinions that cannot be verified. This mix of normative and empirical texts may have impacted how the clusters formed.

Expanding coverage to include diverse policy areas and geographic contexts would improve generalizability. We recommend scaling the dataset and balancing class representation would mitigate skewed learning outcomes and allow the models to develop a more holistic understanding of political language.

6. Conclusion

Key Takeaways

In this project, we tackled detecting political bias in online content by classifying text into left, center, or right ideological categories. Using supervised, machine learning models, Logistic Regression, DNNs, and DistilBERT, we measured their performance across precision, accuracy, F1, and AUC-ROC metrics. Through this, we found that our lowest performance comes from Logistic Regression + TF-IDF, our second best from DNNs, and our best from DistilBERT. For example, our validation accuracy, which started at Logistic Regression's 22% increased to DNN's 27% went to DistilBERT's 42%. Additionally, the confusion matrix shows a steady increase in true positives throughout the progression of our three models. Additionally, there is also a small increase in the AUC-ROC between the models. DistilBERT is a clear step up from TF-IDF + Logistic Regression and DNN. However, it has struggles, especially with the "center" class. Overall, we've trained a model that is good at recognizing strongly biased language especially in left and right leaning texts but has weaknesses in more nuanced and centrist data.

We also learnt through unsupervised learning, like clustering with k-means and hDBSCAN, we were able to identify the natural groupings of text embeddings with the implementation of SimCSE. Although SimCSE embeddings created meaningful semantic representations, it was limited in its ability to capture semantic patterns within clusters based on political bias (although quite effective for misinformation data). Furthermore, unsupervised learning was mostly used for exploratory purposes and is not definitive. so it must be interpreted carefully and not used as a precise measurement.

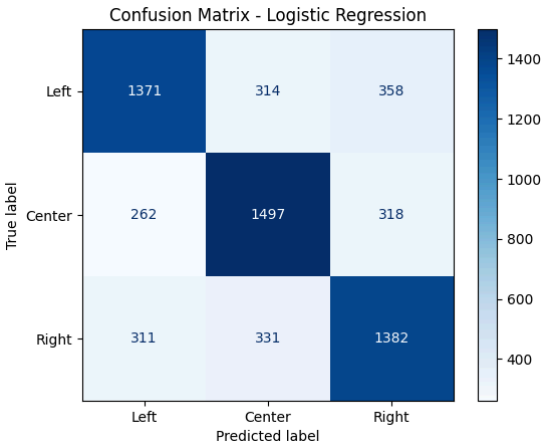
References

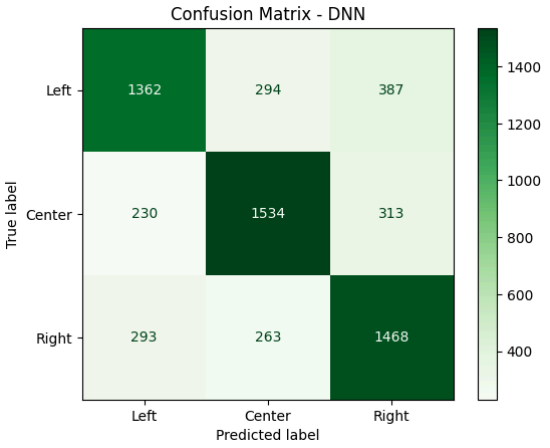
Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*. Available at: https://huggingface.co/distilbert-base-uncased

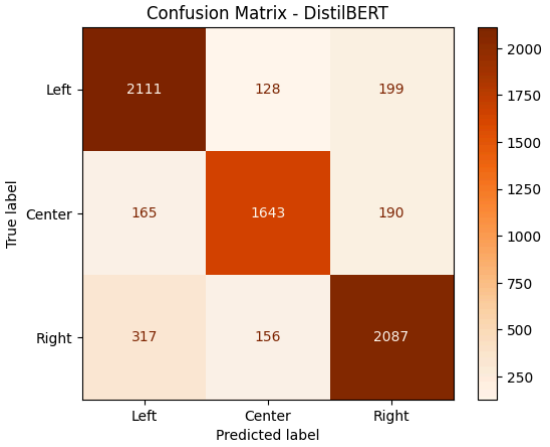
Siddharth Menon. (2023). *Political Bias Prediction Dataset - Media Splits*. Available at: https://huggingface.co/datasets/siddharthmb/article-bias-prediction-media-splits

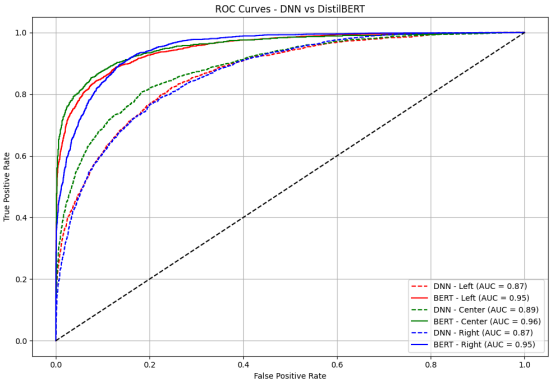
Roupen Minassian. (2023). *Twitter Misinformation Dataset*. Available at: https://huggingface.co/datasets/roupenminassian/twitter-misinformation

Gao, T., Yao, X., & Chen, D. (2021). SimCSE: Simple Contrastive Learning of Sentence Embeddings. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Available at: https://github.com/princeton-nlp/SimCSE









📊 DistilBERT Results:

	precision	recall	f1-score	support
Left	0.81	0.87	0.84	2438
Center	0.85	0.82	0.84	1998
Right	0.84	0.82	0.83	2560
accuracy			0.83	6996
macro avg	0.84	0.83	0.84	6996
weighted avg	0.84	0.83	0.83	6996

📊 Logistic Regression Results:

	precision	recall	f1-score	support
Left	0.71	0.67	0.69	2043
Center	0.70	0.72	0.71	2077
Right	0.67	0.68	0.68	2024
accuracy			0.69	6144
macro avg	0.69	0.69	0.69	6144
weighted avg	0.69	0.69	0.69	6144

Epoch 1/10
768/768
━━━━━━━━━━━━━━━━━━━━
━━━ 7s 4ms/step - accuracy: 0.5257 - loss: 0.9555 - val_accuracy: 0.6914 - val_loss: 0.7040
Epoch 2/10
768/768
━━━━━━━━━━━━━━━━━━━━
━━━ 3s 3ms/step - accuracy: 0.7646 - loss: 0.5837 - val_accuracy: 0.7103 - val_loss: 0.6839
Epoch 3/10

768/768
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━
━━━ 2s 3ms/step - accuracy: 0.8343 - loss: 0.4387 - val_accuracy: 0.7121 - val_loss: 0.7301
Epoch 4/10
768/768
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━
━━━ 5s 6ms/step - accuracy: 0.9044 - loss: 0.2897 - val_accuracy: 0.7186 - val_loss: 0.7793
<keras.src.callbacks.history.History at 0x7abe2ae90b90>

confusion matrix - logistic regression:
predict left, true left = 1371;

Predict left, true center = 262;

Predict left, true right = 311

Predict center, true right = 311

Predict center, true center = 1497

Predict center, true left = 314

Predict right, true right = 1382

Predict right, true center = 318

Predict right, true left = 358

confusion matrix - DNN:

predict left, true left = 1362;

Predict left, true center = 230;

Predict left, true right = 293

Predict center, true right = 263

Predict center, true center = 1534

Predict center, true left = 294

Predict right, true right = 1468

Predict right, true center = 313

Predict right, true left = 387

confusion matrix - Distilbert:

predict left, true left = 2111;

Predict left, true center = 165;

Predict left, true right = 317;

Predict center, true right = 156

Predict center, true center = 1643

Predict center, true left = 128

Predict right, true right = 2087

Predict right, true center = 190

Predict right, true left = 199