| **COMP 590-158 Final Project Report** |
| --- |
|  |
| **Partisanship on US News Sites** |
| Jahnavi Alapati - 730393699 - [jalapati@unc.edu](mailto:jalapati@unc.edu)  Maya Krishnamoorthy - 730391571 - [mkrishn@email.unc.edu](mailto:mkrishn@email.unc.edu) |
|  |
|  |
|  |

**Abstract**

Our study aims to investigate the political biases in U.S. news organizations through machine learning models and understand the impact of these biases on the public. Our original dataset contained 21,005 articles from 13 U.S. news organizations alongside their political leanings, rated by judges through Amazon's Mechanical Turk. Our models use feature extraction techniques like TF-IDF and Bag of Words on logistic regression, Naive Bayes, SVM, and an LSTM neural network. The results show similar performance in their evaluation metrics, with LSTM performing the best. These results highlight the limitations of working with textual data and hidden political nuances. In the future, we will continue to explore advanced models that better capture and recognize political language.

1. **Motivation and Problem Definition**

Partisanship is a dedication or loyalty to a political party, usually accompanied by a negative view of the opposing party. Extreme partisanship is generally regarded as detrimental to the functioning of the government because of its ability to misrepresent political reality. News organizations have extensive power over the public nowadays due to their mass production of information. They have the ability to introduce polarizing conversations that affect how the public feels about political issues. The goal of this project is to investigate and identify the pre-existing bias in new organizations concerning political events worldwide. We aim to provide transparency about the publications' political alignment and address the influence news organizations have over public perception. Our goal is to empower the public to analyze their political ideologies more independently.

1. **Methods**

We intend to train a model to take in a news article and output a label describing the political leaning of the article (Democratic, Republican, or Neutral). We used articles from the Quantifying News Media Bias through Crowdsourcing and Machine Learning Dataset published by the University of Michigan to train our model. The variables of interest are the article’s URL and the judge’s vote on how the articles lean politically toward the Democratic, Neutral, or Republican parties. The original dataset consists of 21,005 articles from 13 news organizations. Judges, sourced from Amazon’s Mechanical Turk, qualified the political attitude of each article toward the Democratic and Republican parties as Positive, Somewhat Positive, Neutral, Somewhat Negative, or Negative after reading its content.

Our first step was to map a supervised label for our models using the judge’s scores for the political attitudes of the articles. We began by mapping the qualitative scores of political attitudes to their quantitative counterpart by setting up a mapping equation that respectively assigned the values to [1, 0.5, 0, -0.5, -1]. We took the minimum of both values to create a “negatively\_affected” column that described which party was more negatively affected by the article’s tone. Whichever value was lower would be mapped to either “Democrat”, “Republican”, or “Neutral.” We then took the opposite of those values, where the opposite of “Democrat” was “Republican,” to create the “leaning\_toward” variable, which referenced which party the article leaned towards.

Since the dataset only gave us the URL, we scraped the content of the article to use as our models’ input. We created a “text\_content” column that scraped the article's text from the URL and then normalized the text to remove redundancies, improve the integrity, and allow for the models to predict better. Our normalization techniques included stopword removal, special character removal, number removal, tokenization, lemmatization, and NER tagging. We specifically used NER tagging because of the sensitivity of our articles to organizations, people, and acronyms. Lastly, we also created an organization column that will tell us what publisher released the article.

Post URL-scraping, we found that many articles were blocked by a paywall, specifically from the Wall Street Journal, New York Times, and Chicago Tribune. All articles blocked by a paywall or pop-up were discarded from the dataset for model training and testing. Ultimately, we had 10,776 usable articles for evaluation. Our final dataset consisted of the original 7 variables (URL, news type, perceived, primary topic, secondary topic, Democratic vote, Republican vote), along with the normalized text content, party approval (leaning\_toward), party disapproval (negatively\_affected), and the publishing organization.

**A graph of a column

Description automatically generated with medium confidence**

Figure 1: Distribution of Scraped Articles by News Organization

Finally, we used TF-IDF and Bag of Words as our main methods of feature extraction for our models. TF-IDF was used because of the ability to weigh the frequency of tokens across articles. BoW was used because of its ability to measure the frequency of word occurrence across articles as a whole.

1. **Experiments and Results**

To avoid overfitting our models and ensure they are trained properly, we first split our scraped and normalized dataset into an 80%/20% train/test split. We began by training a baseline random classifier with all of the usable articles in our dataset. The baseline classifier gave us an accuracy of approximately 40%, where the number of neutral articles was thrice as much as the articles for either party. As the random classifier was stratified, we saw that the precision of the Neutral articles was higher than the precision of the Democratic and Republican articles, which skewed the accuracy of the model. To prevent our models from training too heavily on neutral articles, we decided to address this by balancing our data and randomly sampling from the Democratic (2,435) and Neutral (6,212) leaning articles so that they were equal to the number of Republican-leaning articles (2,129). After running a random classifier on the resampled dataset, we found an overall accuracy of 32%, which aligned with a random distribution of three labels.

A graph of different sizes and numbers

Description automatically generated with medium confidence

Figure 2: Distribution of Articles by Political Leaning (pre and post-Sampling)

A table with numbers and a number of numbers

Description automatically generated with medium confidence

Figure 3: Random Classifier Performance Table

After running the baseline models, we began to train other basic models, such as logistic regression, Naive Bayes, SVMs, and LSTM, on our balanced dataset. We found that each model has its strengths and weaknesses, each working differently to classify articles according to their political leanings. We used both Term Frequency-Inverse Document Frequency (TF-IDF) and Bag of Words (BoW) to extract our features from each of the articles.

Overall, we found that our models performed slightly better with TF-IDF, but our accuracy generally ranged between 55% and 59% for all models. Models with TF-IDF generally performed better over BoW because it is evaluated not just by how often it appears in a certain document, but also in the entire dataset, allowing it to indicate political leanings better. Logistic regression and SVM performed better than Naive Bayes in every sense. Naive Bayes assumes that features are independent of each other given the class label, whereas logistic regression and SVM do not make this independence assumption and are capable of capturing interactions between features. Moreover, Naive Bayes directly estimates probabilities from the data, rather than the optimization techniques used in Logistic Regression and SVM, which may have led to better performance from the latter.

We can see that the models generally achieved higher precision for Republican-leaning articles, suggesting that when the model predicts an article is Republican-leaning, it is very likely to be correct. This can indicate that the defining features or keywords for Republican-leaning articles are more distinctive or consistently used within these articles. The models also achieved higher recall for Democratic-leaning articles, but lower precision. This combination implies that the model results in a significant number of false positives. We also noticed the general trends for precision, recall, and F1 score are related to how few of the total articles of each category were sampled – there are lower scores for the labels that were less sampled.

A table with numbers and symbols

Description automatically generated

Figure 4: Logistic Regression, Naive Bayes, and SVM Model Performance

We also compared these basic models to a Long-Short Term Memory (LSTM) neural network. This model performed better than all the other models, with an overall accuracy of 61%. Across epochs, there's a steady decrease in loss and an increase in accuracy for both training (loss from 1.03 to 0.82 and accuracy from 0.52 to 0.63) and validation sets (validation loss from 1.07 to 1.03 and validation accuracy from 0.48 to 0.61), suggesting that the model is learning and improving its performance on the task with each epoch. The relatively close loss values and slight accuracy differences between training and validation suggest that the model is not overfitting and is generalizing reasonably well to unseen data. As LSTMs are recurrent, they are well-suited for text data. They can process documents of varying lengths and retain the phrases or terms that significantly influence an article's political leaning. By capturing these nuances, the LSTM can provide a more accurate political orientation classification than the "basic" models.

A table with text and numbers

Description automatically generated

Figure 5: LSTM Performance

After examining model performance, we also chose to look at how words were classified for each label and how strongly they affected both the Logistic Regression model and the Naive Bayes model that used TF-IDF features. In Logistic Regression, higher positive coefficients suggest words strongly indicative of a class, with "republican," "obama," and "said" being the most significant for Democratic, Republican, and Neutral categories, respectively. In contrast, Naive Bayes assigns log probabilities, with lower values (more negative) indicating stronger association. We see that words like "nbc" and "huffpost" appear influential for all three classes, perhaps suggesting these terms are common in the dataset but may not be as effective for differentiation. Similarly, Since Logistic Regression assigns coefficients for features in the context of each separate class, the results are more likely to be sensitive to the political tone of each token.

We also explored which words are least likely to appear in each class, and the results showed us that Logistic Regression can better restrict words for different classes. For example, “republican” has a high coefficient for the Democratic class versus a low coefficient for the Republcan class, meaning it is far less likely for the word “republican” to appear in the Republican versus Democratic class.

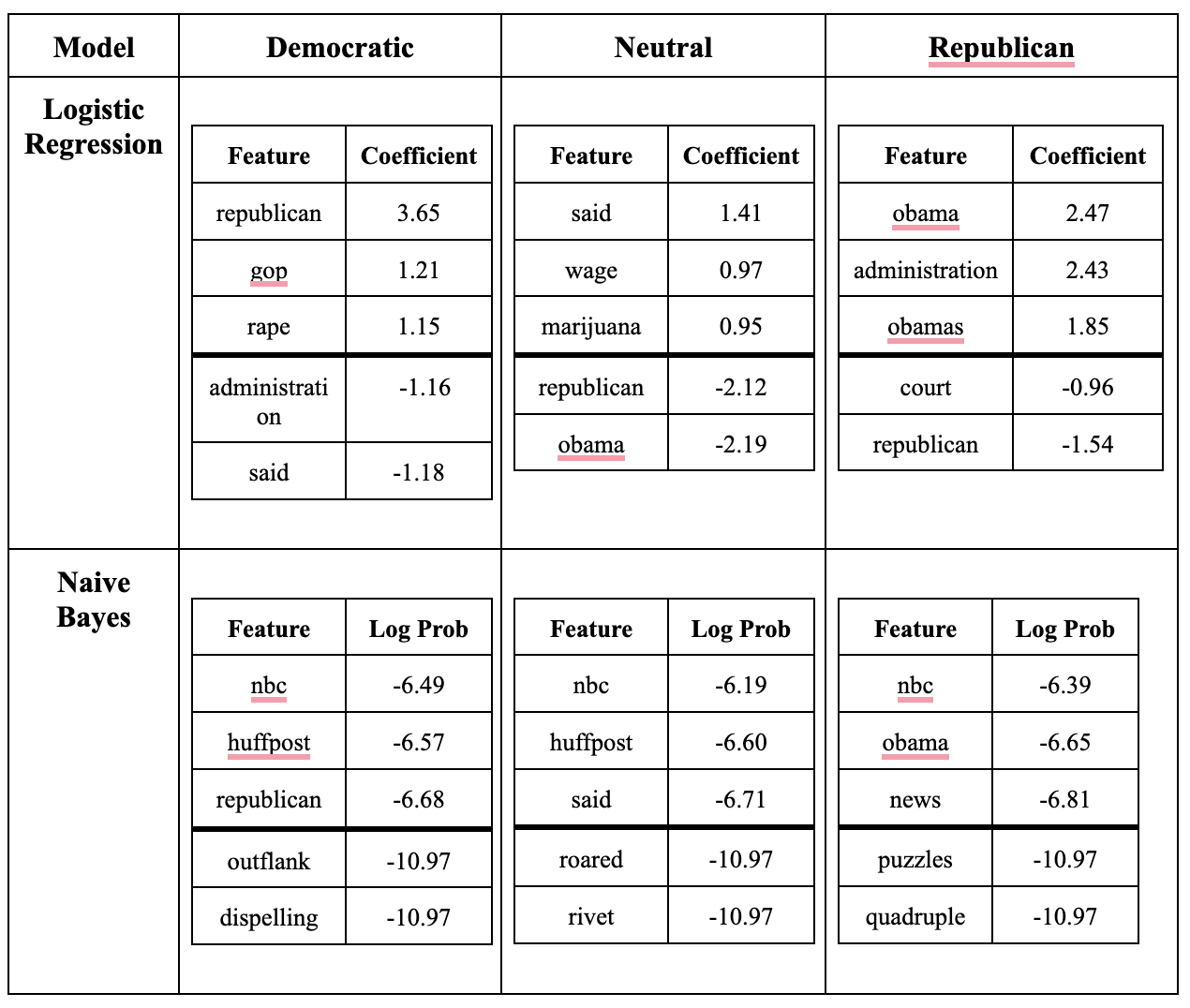


Figure 6: Strongest and Weakest Words of Interest for Each Class in LR and Naïve Bayes Models

**4** **Conclusion**

When evaluating the political leaning of news articles using machine learning techniques, it is important to understand the limitations of the data input. Scraping news articles may not be the best way to gather text content, especially since news organizations implement subscription pop-ups and site blocks to limit site scraping. Moreover, human judging of news articles may have also introduced subjectivity and bias into our dataset and models, resulting in some rows of mislabeled data that conflict with the remaining data.

Overall, we found that our models were not incredibly accurate, as a result of the limitations discussed, however, we have a deeper understanding of the subtle political nuances that publishers use in their articles and the complexity of text data. We hope to test more models, such as BERT or RoBERTa, or transformers, which may be able to provide a more nuanced understanding of word relationships and sentence structures than traditional models like Naive Bayes or Logistic Regression.

**5 Diversity**

Both members of this project, Jahnavi and Maya, started in Computer Science, but have since diverged in their interests. Jahnavi is beginning to transition into the field of finance and looks toward popular U.S. news publications to support her interest and understanding of the sector. Moving forward, Jahnavi hopes to use the knowledge she gained in this course to engage in machine learning projects in her new job and is curious to explore the sentiment of financial reporting by the media.

Maya, a double major in Computer Science and Biostatistics, has been interested in the political effects of healthcare as written in the media. In the fall, Maya is set to start her master’s degree in Health Data Science, where she hopes to explore NLP and machine learning techniques in the healthcare setting. She is especially interested in exploring the political misinterpretations in healthcare reporting. This project represents a culmination of our interests and how news agencies are reporting on the fields of our interest. We are in a unique position to understand different parts of the U.S. news and recognize any political nuances that are in published articles and speak on the implications of public understanding.

**6 Individual Contributions**

For this research project, Jahnavi and Maya both worked together a few times a week to train, test, and evaluate models. They collaborated on designing the overall methodology and strategy and then decided which models to employ. They worked together to source the dataset, scrape the data, and apply feature extraction techniques to the models. Jahnavi and Maya also each ran the models on their own time, checking in with each other whenever they ran into a problem so they could schedule a time to correct it. They regularly reviewed each other's work, provided feedback, and made adjustments as necessary.

**References**

Budak, Ceren, Sharad Goel, and Justin M. Rao. n.d. “Quantifying News Media Bias through Crowdsourcing and Machine Learning Dataset.” Deepblue.lib.umich.edu. Accessed April 28, 2024. <https://deepblue.lib.umich.edu/data/concern/data_sets/8w32r569d>.

**Appendix**

Colab Notebook: <https://colab.research.google.com/drive/1Dt0ALzNn3HwXr_Kc3N4nN4PMpfqz2vVX?usp=sharing>