Danmarks Tekniske Universitet

DTU

Republican or Democrat? A Model-Based Machine Learning Approach for Classifying Tweets

Authors

Group 15:
Mads Yar - s193992
Salik Muneeb - s215133
Ignacio Ripoll - s242875
Carlos Fernandez Liger - s243308

Course:

42186 Model-Based Machine Learning

Hand-in:

May 29, 2025

Contents

Abstract

This project looks into probabilistic model-based machine learning approaches for classifying political tweets from the 2020 U.S. presidential election. We implement and evaluate two Bayesian models: a simpler Bayesian Logistic Regression (BLR) and our own adaptation of a more complex Bayesian Hierarchical Model (BHM). Both models try to distinguish between Democratic and Republican tweets based on text and other features. Our goal is to test whether a more complex model, which includes topic modeling and clustering, leads to better classifications compared to a classic binary Bayesian model.

1 Introduction

The 2020 U.S. presidential election was one of the most discussed events of that year. It attracted large media coverage, public interest and international attention. Since it was a trending topic, Twitter saw increased activity related to both the Democratic and Republican parties and their candidates. Tweets were posted for and against both sides [1]. This sparked many different opinions which we aim to analyze in this project. By implementing two probabilistic models, we try to classify the political affiliation behind tweets. In other words: Democratic vs. Republican. The goal is to investigate whether model-based machine learning approaches can effectively distinguish between these two classes based on the tweet text and associated features. To perform this classification, we use the following two Bayesian models:

- Bayesian Logistic Regression (BLR)
- Bayesian Hierarchical Model (BHM)

In this project, BLR will act as a simple baseline since BLR is commonly used for probabilistic classification tasks [2]. In contrast, the BHM is a more flexible and allows for other probabilistic components to be included. To follow the model-driven project description, we adapted a traditional BHM and create our own version or model by combining several modeling ideas covered during the course. To be more precise, we incorporated two additional probabilistic sub-models into our own BHM:

- **Latent Dirichlet Allocation (LDA):** is used for topic modeling to try to uncover latent semantic topics across tweets [2].
- **Gaussian Mixture Model (GMM):** is used to group tweets by length, punctuation, and tone to help the model spot different ways people express political opinions. [3]

These components are combined with sentiment scores (from VADER [4]) and TF-IDF [5] to allow the model to hopefully capture richer interactions between text structure, meaning and tone. This has been done to see if it is possible to better capture the political discourse on Twitter. By building our own BHM ourselves and implementing it in Pyro, we follow the model-driven aspect of the project. Thereby, testing whether a more expressive model leads to more balanced and accurate classification.

Once the models are implemented, they are evaluated and compared. We analyze training and validation loss and accuracy to assess generalization. Confusion matrices and ROC curves are also included to understand class balance and classification performance under uncertainty. In the following sections, we will provide a detailed breakdown of each model and hopefully conclude which model performs the best.

2 Data Processing

The dataset used for this project consists of tweets related to the 2020 U.S. presidential election. It is available to download on Kaggle [6]. It contains two separate CSV files. One CSV contains tweets associated with Donald Trump while the other contains tweets about Joe Biden. Each tweet in the dataset also includes metadata such as the state from which it originated, amount of likes, retweets and the time of posting. The combined dataset initially consisted of 1,747,805 tweets. Several pre-processing steps were applied to clean and refine the dataset for better classification:

1. **Combining Datasets:** The two CSV files were then merged into a single dataframe, and duplicate tweets were removed to avoid redundancy.
2. **Text Cleaning:** Tweets were cleaned by doing the following:
    - Removing URLs, mentions, emojis, and non-alphabetic characters.
    - All text was lower-cased.
    - Excessive whitespace was removed.
    - Hashtags were split into actual words to preserve semantic meaning. For example, #Biden2020 becomes Biden 2020 ensuring that hashtags are treated as natural language components during analysis.
3. **Candidate Labeling:** Tweets were labeled as either Republican (1) or Democrat (0) based on if they mentioned Trump or Biden using a list of common names, nicknames, and political terms. Tweets that mentioned both or neither were excluded.
4. **Feature Extraction:** Sentiment scores were extracted using VADER along with statistical features like tweet length and punctuation count. Compound sentiment was also combined with topic distributions to capture sentiment-topic interactions.
5. **Vectorization and Topic Modeling:** Common political terms were removed using a custom stop word list to reduce noise. TF-IDF captured important words, while LDA was used to extract key topics to help the model learn better from the tweets.

After cleaning all the tweets, the dataset was balanced by randomly sampling an equal amount of tweets from each political party. They were distributed as follows:

- **Trump-only mentions:** 600,000 tweets.

- **Biden-only mentions:** 600,000 tweets.
- **Mentions of either candidate:** 1,200,000 tweets.

3 Models

In this project, two probabilistic models have been implemented with the purpose to classify political tweets as either Republican or Democratic. The first model is a standard Bayesian Logistic Regression model, as mentioned in the introduction. It was chosen to serve as a baseline when comparing to our custom model. The second model is a custom-adaptation of a Bayesian Hierarchical Model (BHM). The model was developed to see whether it could learn more effectively from the tweets by capturing more complex relationships between features like sentiment, topics, and text characteristics. Lastly, these choices were grounded in the principles of Model-Based Machine Learning (MBML), which prioritizes building interpretable models tailored to the uncertainty of the problem at hand [7].

3.1 Bayesian Logistic Regression (BLR)

Bayesian Logistic Regression is an extension of logistic regression for binary classification [2]. It allows the model to treat its parameters as random variables with prior distributions to provide uncertainty estimations and regularization [2]. Each binary label $y_n \in \{0, 1\}$ is drawn from a Bernoulli distribution whose probability is governed by a sigmoid transformation of a linear combination of input features::

$$p(y_n \mid \mathbf{x}_n, \boldsymbol{\beta}, b) = \text{Bernoulli}(\sigma(\boldsymbol{\beta}^\top \mathbf{x}_n + b))$$

where $\mathbf{x}_n$ is the feature vector of the nth tweet, $\boldsymbol{\beta}$ is the weight vector, b is the bias term, and σ is the sigmoid function:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

The model parameters $\boldsymbol{\beta}$ and b are treated as latent variables with Gaussian priors [2]:

$$\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad b \sim \mathcal{N}(0, 1)$$

This model assumes that the log-odds of the class label can be expressed as a linear function of the input features. While this makes the model interpretable and computationally efficient, it also limits its ability to capture more complex patterns in the data. To improve flexibility, a bias term b is included.

A Bayesian approach is used by placing Gaussian priors over both the weights and the bias term. These priors help regularize the model by encoding uncertainty about the parameters and reducing the risk of overfitting, especially in high-dimensional spaces. The posterior distribution over the parameters is approximated using variational inference.

To show the full structure of the model, the generative process is outlined in Figure 1.

1. Draw weights $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2. Draw bias $b \sim \mathcal{N}(0, 1)$
3. For each tweet $n = 1, \ldots, N$:
   (a) Compute logits: $\ell_n = \boldsymbol{\beta}^\top \mathbf{x}_n + b$
   (b) Compute probability: $p_n = \sigma(\ell_n)$
   (c) Draw label: $y_n \sim \text{Bernoulli}(p_n)$

Figure 1: The generative process for our BLR.

Which is illustrated by the graphical model in Figure 2.

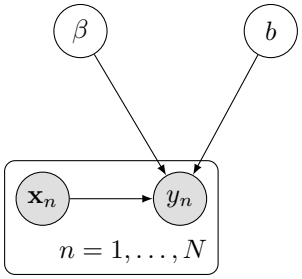

Figure 2: The probabilistic graphical model for our BLR.

3.2 Bayesian Hierarchical Model (BHM)

To better model the nuanced structure of tweets, we developed our own Bayesian Hierarchical Model that integrates several features under one coherent probabilistic machine learning framework. It was developed to see whether it could learn more effectively from the tweets by modeling complex relationships between multiple types of features. The goal is to improve classification by modeling how features like sentiment, topics, and text characteristics interact instead of treating them independently.

The model combines two other sub-models, which are a Latent Dirichlet Allocation (LDA) and a Gaussian Mixture Model (GMM) within our BHM. This has been done to try structuring the latent space of these tweet features. The LDA is used to capture the semantic topics in the tweets, while the GMM is used for modeling clusters in the topic-sentiment space. These were included to help the model understand both the meaning of the tweets and how people feel about those topics, which is important for making more accurate classifications. Additionally, the sub-models are combined with sentiment scores and other statistical text features. The model also captures dependencies between these components such as how sentiment may be different depending on the topic. This allows the model to learn patterns that a simpler model like BLR cannot. The main features used in the model are:

- **Topic distributions (LDA):** Capture the semantics in each tweet.
- **Sentiment:** Quantify the tone using positive, negative, neutral, and compound scores.
- **Stat.- and TF-IDF features:** Describe text characteristics and lexical content.
- **GMM Clustering:** Models latent structure in the topic-sentiment space.
- **Cross-feature dependencies:** Sentiment-topic and topic-feature interactions.

We use a global scale parameter $\tau \sim \text{HalfCauchy}(1.0)$ over all of the model's weights which uses Gaussian priors [8]. This has been done to ensure regularization in a principled way [8]. Additionally, it controls how much each component is allowed to influence the model. Hence, preventing a single parameter from overtaking the learning. All features (topics, sentiment, clusters, TF-IDF and text statistics) are included in the final logit used for classification. This structure allows the model to integrate different types of information into a single decision while still reflecting the uncertainty in each part. To better show our model, the generative process is outlined in Figure 3.

1. Draw global scale $\tau \sim \text{HalfCauchy}(1.0)$
2. Draw GMM mixture weights $\boldsymbol{\pi} \sim \text{Dirichlet}(\mathbf{1})$
3. For each cluster $k = 1, \ldots, K$:
   (a) Draw mean: $\boldsymbol{\mu}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
   (b) Draw std dev: $\boldsymbol{\sigma}_k \sim \text{LogNormal}(0, 0.5)$
4. Draw model weights:
   (a) Topic weights $\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \tau^2\mathbf{I})$
   (b) Sentiment-topic weights $\boldsymbol{\Phi} \sim \mathcal{N}(\mathbf{0}, \tau^2\mathbf{I})$
   (c) Text statistics weights $\boldsymbol{\gamma} \sim \mathcal{N}(\mathbf{0}, \tau^2\mathbf{I})$
   (d) TF-IDF weights $\boldsymbol{\psi} \sim \mathcal{N}(\mathbf{0}, \tau^2\mathbf{I})$
5. For each tweet $n = 1, \ldots, N$:
   (a) Draw cluster: $z_n \sim \text{Categorical}(\boldsymbol{\pi})$
   (b) Draw GMM features: $\mathbf{g}_n \sim \mathcal{N}(\boldsymbol{\mu}_{z_n}, \boldsymbol{\sigma}^2_{z_n})$
   (c) Compute final logit:

$$\text{logits}_n = \sum_i g_{ni} + \boldsymbol{\theta}^\top \mathbf{t}_n + \mathbf{s}_n^\top \boldsymbol{\Phi} \boldsymbol{\theta} + \boldsymbol{\gamma}^\top \mathbf{f}_n + \boldsymbol{\psi}^\top \mathbf{v}_n$$

   (d) Draw label: $y_n \sim \text{Bernoulli}(\sigma(\text{logits}_n))$

Figure 3: The generative process for our Bayesian Hierarchical Model.

Which results in the graphical model shown in Figure 4.

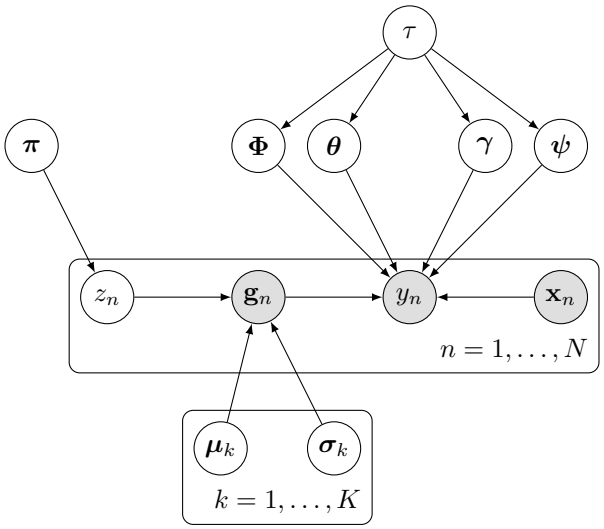

Figure 4: PGM for the Bayesian Hierarchical Model

Although $\mathbf{g}_n$ is derived from components of $\mathbf{x}_n$ during preprocessing, in the probabilistic model $\mathbf{g}_n$ is treated as an observed variable generated from the latent variables z_n, $\boldsymbol{\mu}_k$, and $\boldsymbol{\sigma}_k$. Since $\mathbf{x}_n$ does not probabilistically influence $\mathbf{g}_n$ within the model definition, no arrow should point from $\mathbf{x}_n$ to $\mathbf{g}_n$ in the graphical model.

4 Evaluation

In this section, our two models will be evaluated. A range of different evaluation metrics have been used to measure each model's performance. To be more specific, the following metrics have been used: Accuracy, loss, area under the curve (AUC), confusion matrices and ROC-curves. The goal is to understand how well each model can classify tweets as either Democratic or Republican. Additionally, this is also to see if either model is biased towards a specific class which in this case is Democratic or Republican. Lastly, we visualize training loss and accuracy to examine convergence and generalization. To give a quick overview model comparison, table 1 has been created.

Model	**Accuracy (%)**	**AUC**	**Train Time**	**Notes**
BLR	61.3	0.51	ca. 3 min	Biased towards the Democratic class
BHM	61.8	0.65	ca. 82 min	More balanced performance

Table 1: Performance comparison between our models.

From table 1, it can be seen that the BHM achieves slightly better accuracy and a significant higher AUC score than the BLR. This shows that the BHM is better at differentiating between our two classes. However, this comes at the cost of a much longer training time.

4.1 Accuracy and Loss

Figures 5 and 6 show the training and validation loss and accuracy curves for each model. Here, the left plot, in figures 5 and 6, showcases the loss for both the training (blue line) and for the validation (red line). Whereas the right plot, in both 5 and 6, shows the training accuracy (blue line) and the validation accuracy (green line).

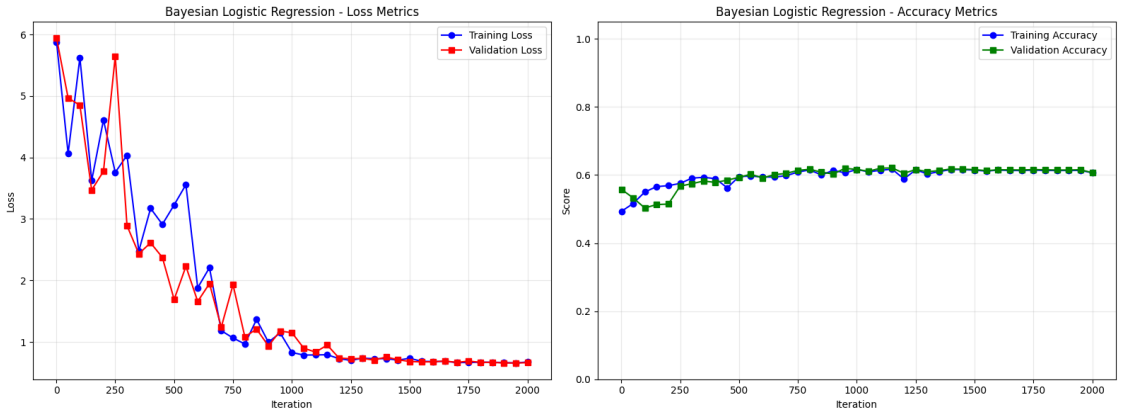

Figure 5: Training metrics for Bayesian Logistic Regression (loss and accuracy).

The BLR converges quickly and remains stable during training. Both loss curves decreases smoothly during the iterations. Likewise, both the validation- and training accuracy increases slowly, capping out at around 0.61. Lastly, the validation accuracy remains close to the training curve, which shows the model has good generalization and minimal overfitting.

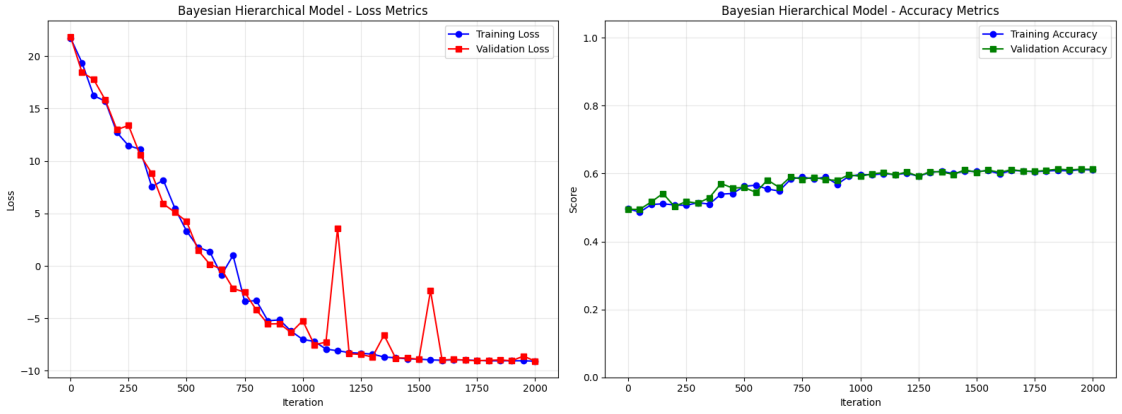

Figure 6: Training metrics for Bayesian Hierarchical Model (loss and accuracy).

In contrast, the BHM shows a more slow learning curve. Despite the spikes in the validation loss, the model recovers quickly and ultimately reaches slightly higher validation accuracy compared to the BLR. Additionally, the BHM accuracy curves are trending more upward toward the end compared to the BLR curves which remains more flat towards the end. This trend suggests that the BHM may benefits from more training iterations. Despite being more complex, the model maintains stable learning and converges well.

4.2 Confusion Matrix Analysis

Figure 7 shows the confusion matrices for both models. The BLR model correctly predicts a high number of Democratic tweets (80,053), but it misclassifies a large portion of Republican tweets (53,001). This shows a clear class imbalance. The BHM, on the other hand, achieves a slightly more balanced distribution of predictions: 68,523 correctly classified Democrats and 79,819 correctly classified Republicans. This highlights the BHM's ability to generalize better across classes.

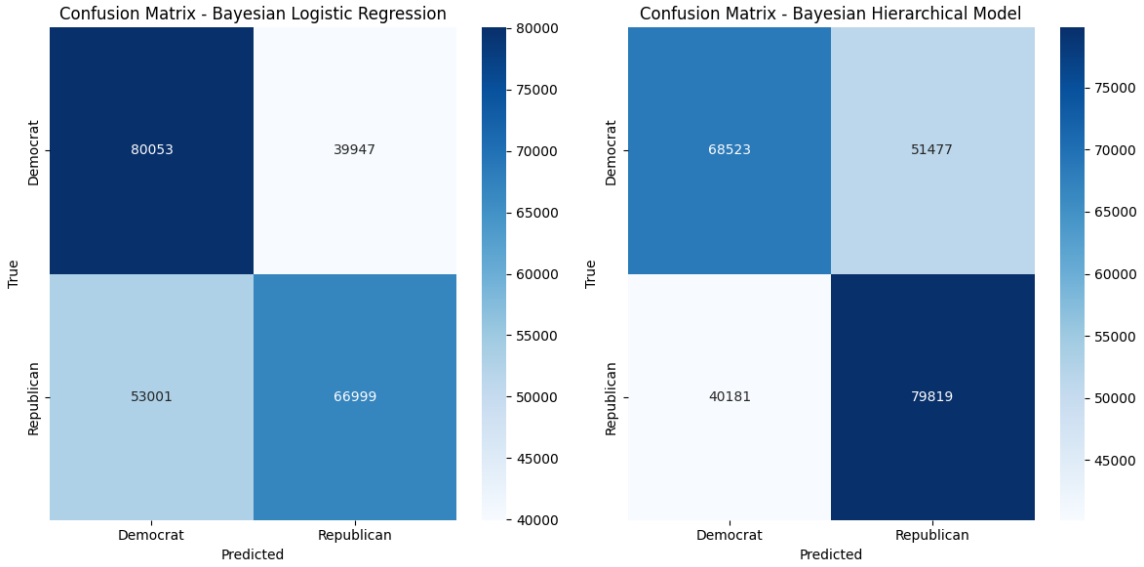

Figure 7: Confusion matrices for BLR (left) and BHM (right).

To further break down performance on the Republican class, the following table shows the true positive rate (TPR) and false positive rate (FPR):

Metric	BLR	BHM
True Positive Rate	0.558	0.665
False Positive Rate	0.332	0.429

Table 2: TPR vs FPR for the Republican class.

The BHM achieves a significantly higher TPR which shows it captures Republican tweets more accurately. However, it also has a higher FPR which means it misclassifies more Democratic tweets as Republican. This trade-off reflects the BHM's complexity because it captures subtle patterns better but may overfit in ambiguous cases. Examples of ambiguous cases could be using generic language such as "We need real change in Washington". This could be about either party. Other cases could be sarcasm, humor or conflicting sentiment (like "Biden did okay, but we need a harder immigration policy"). Meanwhile, the BLR is more conservative with fewer false positives but also misses more Republican tweets.

4.3 ROC Curve Analysis

The ROC curve shows a model's ability to choose between classes by plotting the true positive rate against the false positive rate. A higher AUC means the model can better rank tweets by likelihood of being Republican or Democratic. Looking at figure 8 shows that the BHM achieves a much better ROC curve with an AUC of 0.6539 compared to 0.5123 for BLR. This confirms that the BHM is more effective at separating between Democratic and Republican tweets.

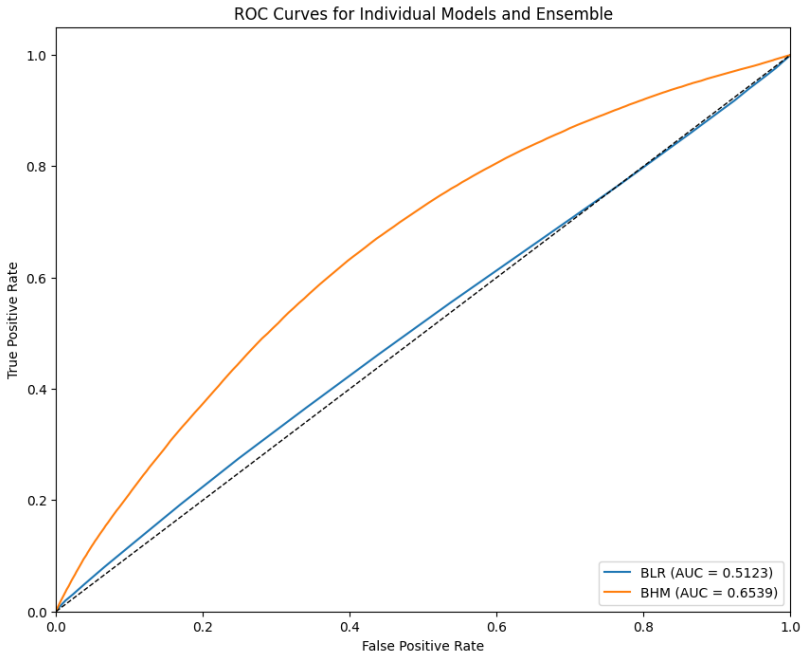

Figure 8: ROC curves for Bayesian Logistic Regression and Bayesian Hierarchical Model.

To conclude, both models reach similar accuracy. However, the BHM is better at balancing and distinguishing between classes compared to our BLR. This is most likely due to our BHM's ability to model structured feature interactions. Hence, making it more suited for a dataset such as tweet data.

5 Conclusion

We implemented and evaluated two Bayesian models for classifying political tweets: a simple Bayesian Linear Regression (BLR) model and our own more complex adaptation of a Bayesian Hierarchical Model (BHM). They were trained on a dataset of tweets from the 2020 U.S. presidential election. While both models achieved similar accuracy, the BHM showed clear improvements in balance and robustness. It achieved a significantly higher AUC and better class separation compared to our BLR. It was very apparent for the Republican tweets. This suggests that the BHM's ability to model feature interactions, like sentiment, topics, and text structure, adds value to the classification.

However, this comes at the cost of increased complexity and a much longer training time. Despite that, the BHM remains more suitable for handling the nuances in tweet data. Overall, this project shows how model-based machine learning approaches can offer interpretable solutions for text classification under uncertainty.

6 Future Works

While our BHM showed better performance than our BLR, there is still room for improvement as seen in our evaluation section. One idea is, of course, to let it run for more iterations to see if this improves accuracy. Another way is to examine how strongly the different features influence each other to see if some features are more important than others.

Another improvement could be to let the model decide for itself how many GMM clusters it needs, instead of us setting a fixed number. This might help it find more natural groupings in the data. We also used Stochastic Variational Inference (SVI) with ELBO loss for training. In the future, it could be interesting to try other inference methods to see if they improve performance. Lastly, we only used a few text features like sentiment and topics. More features could be added, such as information about the user or the time the tweet was posted. This might help the model learn patterns in how political opinions change over time or across different users.

References

[1] P. R. Center, "Differences in how democrats and republicans behave on twitter." https://www.pewresearch.org/politics/2020/10/15/differences-in-how-democrats-and-republicans-behave-on-twitter/, 2020.

[2] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA: MIT Press, 2012.

[3] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[4] C. Hutto and E. Gilbert, "vadersentiment." https://github.com/cjhutto/vaderSentiment, 2014. Accessed on [04/12/2024].

[5] F. e. a. Pedregosa, "Scikit-learn: Machine learning in python," 2011.

[6] Kaggle, "US Election 2020 Tweets." https://www.kaggle.com/datasets/manchunhui/us-election-2020-tweets/data.

[7] J. Winn, C. M. Bishop, T. Diethe, J. Guiver, and Y. Zaykov, *Model-Based Machine Learning*. Routledge, 2019.

[8] N. G. Polson and J. G. Scott, "On the half-cauchy prior for a global scale parameter," 2011.