Predicting Political Leans and Sentiment Analysis based on Reddit Posts

TDDE16 - Gabriel Hult
gabhu204

Abstract

This project investigated how different classifiers could predict the political lean of Reddit posts, depending on what was said and what words were used. The classifiers used for the investigation were Multinomial Naive Bayes, Bernoulli Naive Bayes, Linear SVM and Random Forest. This was done with an initial untouched dataset and an undersampled version of the dataset. Meanwhile, using the top words that impacted the models' predictions, sentiment analysis was conducted using the Naive Bayes classifiers' top words to analyse what sentiment different political leans tended to have. This was done using two third-party libraries, VADER and TextBlob. The results produced show that Random Forest performed the best overall, achieving an F1-score of 84% when predicting liberal political leans for the unbalanced dataset and 73% for the balanced dataset ($0,27\%$ compared to 0.49%). Overall, all models performed above baseline in all regards except for the recall of the Multinomial Naive Bayes when predicting the conservative class using the unbalanced dataset. The sentiment analysis showed that there were e.g. mixed sentiments regarding political figures. It was also found that VADER tended to show the intensity of a sentiment while TextBlob produced a result representing the general sentiment instead.

1 Introduction

This chapter provides an introduction to the project, including motivation, the goals as well as delimitations of the project.

1.1 Motivation

Politics is a sensitive topic for a lot of people for a lot of reasons. One place on the Internet to discuss politics is Reddit. Reddit is a social media and popular network of forum communities where people discuss various topics such as sports, general advice and games [11]. Also, there is a decent amount of traffic where people post and discuss politics. As the Internet grows, more people resort to social media to express their feelings regarding politics to make their voices heard. This statement disregards the factual correctness of what is said since party bias can heavily influence what is posted.

There is interest in analysing what sentiment the posters have depending on their political lean. This is called *Sentiment analysis* (alternatively opinion mining or emotional AI), which is an NLP process of identifying the emotional tone of a given text [7]. There are multiple reasons as to why sentiment analysis concerning politics is interesting. First of all, you can get insight as to what emotional tone-specific biases towards parties have. Furthermore, people analysing the sentiment can get insight into what subjects specifically spark the tone of discussions, such as climate change.

1.2 Goals

Firstly, the project will perform classification, using a set of classifiers which will be trained to predict what political lean a poster has depending on what is said. These classifiers are *Multinomial Naive Bayes*, *Bernoulli Naive Bayes*, *Linear SVM* and *Random Forest*. What words impacted the overall political lean prediction will be extracted and used in a parallel analysis, to determine the sentiment of the posts and pair that sentiment towards specific political leans. Especially words that commonly pair with a specific political lean. The overall goal of the sentiment analysis is to determine what emotional tone posters have when using commonly used words connected to their political lean. This will be done with the two libraries VADER and TextBlob.

Keep in mind that this is done on a limited dataset which can skew the results somewhat. Also, the results do not determine the overall sentiment of people leaning towards specific parties. However, it can show tendencies of what is said depending

on the political lean but should be considered cautiously.

1.3 Delimitations

For this project, due to time constraints and the focus of the project, the models evaluated will not have their hyperparameters tuned to instead allocate time for the sentiment analysis. While tuning the models to better perform on the Reddit posts could be interesting to investigate, the sentiment analysis allows for a deeper discussion.

Another delimitation that will be made is that the sentiment analysis will only be performed with the undersampled version of the Multinomial Naive Bayes as well as the Bernoulli Naive Bayes models. This is because it would significantly avoid redundancy and reduce the abundance of result data, which would clutter the results. Multinomial and Bernoulli were chosen because these models will be more straightforward to implement to receive the top words for each class (liberal and conservative) by utilising their log probabilities. The reason for only using the undersampled models is to see what sentiment result we receive with a more balanced dataset rather than an unbalanced dataset.

A final delimitation is that the project will focus on English posts rather than any post, regardless of the language. This is because to handle multiple languages, more pre-processing work has to be done which goes away from the focus of the project.

2 Theory

The necessary and relevant theory and background will be presented in this chapter.

2.1 Naive Bayes

Naive Bayes classifiers are probability-based approaches and are also part of the supervised learning group of classifiers. It utilises the posterior probability of an event. The goal is to select the probability class with the highest probability $(p(y = c|x))$ of representing the object in question [5]. This is given by:

$$c_{opt} = \underset{c \in C}{\operatorname{argmax}} P(x|y = c)P(y = c) \quad (1)$$

Since Naive Bayes provides a probability distribution, one advantage of Naive Bayes is that we can tell if the prediction made is uncertain or not. If the prediction is uncertain, we can refuse to classify the data using the classifier [10].

2.1.1 Multinomial Naive Bayes

Multinomial Naive Bayes is one sub-approach to Naive Bayes and is one of the more popular Naive Bayes classifiers used. It works by letting $x_i \in \{1, \ldots, K\}$, having emission probabilities $\theta_1, \ldots, \theta_K$., then the probability of an event x occurring if θ is given by:

$$P(x|\theta) = \frac{n!}{x_1!...x_K!} \prod_{i=1}^{K} \theta_i^{xi} \quad (2)$$

where the underlying assumption of the classifier is a multinomial distribution of features [5].

2.1.2 Bernoulli Naive Bayes

The approach of Bernoulli Naive Bayes is a classifier which represents documents as a binary vector in a given space. If we consider a binary variable $x \in [0, 1]$ where we define θ as the probability of the variable's occurrence [5], then it is given by:

$$P(x|\theta) = \begin{cases} \theta & \text{if x=1} \\ 1 - \theta & \text{if x=0} \end{cases} \quad (3)$$

2.2 Linear SVM

Support Vector Machines (SVMs) are a collection of supervised learning methods which can be used for regression, classification and outlier detection problems. A support vector machine constructs a set of hyperplanes in a high-dimensional feature space. In many real-world cases, problems are not linear separable and are typically non-linear separable instead. We can map the non-linear problem to a linear problem, then map the linear problems to a high-dimensional feature space using a non-linear function $\phi(x)$. This feature space can then be used to create an optimised hyperplane [15].

2.3 Decision Tree

Decision tree is a supervised learning approach used for regression and classification problems. It is commonly used to build models that closely resemble human reasoning [8]. It is a hierarchical tree structure, consisting of a root node, branches, internal nodes and leaf nodes. The leaf nodes represent every possible outcome for a dataset. Decision trees utilise a divide and conquer strategy by utilising greedy search to find optimal split points within a tree. This is done recursively until the majority of records in the dataset have been classified with a

specific label. Large decision trees tend to lead to overfitting, because of the loss of pure leaf nodes, as too little data falls within a given subtree [4].

2.4 Random Forest

Random Forest classification is a supervised learning approach used in both regression and classification problems [6]. For classification, Random Forest is a meta-estimator which fits multiple decision trees on different sub-samples of the given dataset, while also using averaging to improve the predictive accuracy and control the risk of overfitting [12]. This is one of the advantages of using the Random Forest classifier rather than the Decision Tree classifier, also known as *bootstrapping* [16]. The steps involved when using the Random Forest classifier:

1. Divide the training set into subsets for each of the decision trees
2. Create individual decision trees for each subset from the training set
3. Each decision tree produces a prediction
4. Majority voting is chosen which is the final prediction from the model

2.5 VADER

Valence Aware Dictionary and sEntiment Reasoner is a classifier which is designed to analyse the sentiment of text. VADER is a rule-based approach which focuses on performing well on text styles commonly found on social media. Its sentiment lexicon is considered *gold-standard* after human evaluation [3].

2.6 TextBlob

TextBlob is a library for processing textual data. It comes with a lot of features such as its implementation of Naive Bayes and Decision Tree classification, tokenisation and more features similar to spaCy. It also includes a sentiment analyser. There are two types of sentiment analysers for TextBlob. The first approach is a Naive Bayes approach while the second approach is a pattern analyser [9].

3 Data

This section presents the used dataset. Furthermore, how the pre-processing was done will also be covered.

3.1 Dataset

For this project, a single dataset created by Neel Gajare [1] containing 13000 Reddit posts was used to train, validate and test the sentiment analyser. 65% out of all the posts were labelled as liberal, so an attempt using the original data will be performed as well as an attempt at undersampling for comparison. The dataset contains multiple columns such as *Title*, *Text* and more. These two columns were specifically selected to classify what post belonged to what political lean. Whether the models predict correctly will be validated by comparing the prediction to the actual labelled answer, which is stored in the *Politcal Lean* column of the dataset. There were other columns such as *Subreddit* which is where it was posted. The specific subreddits were not used to be unbiased regarding where the post originated from. Liberals and conservatives likely post on their subreddits which would heavily skew the predictions when it is instead sought after to predict based on the title and text alone.

3.2 Pre-processing

To properly use the data to predict what political lean a poster had and its sentiment, the data had to be pre-processed to mitigate the risk of noise for the trained model, leading to better-trained models. For these classifiers, the pre-processing was done using steps 1, 2.1, 3, 4, 5 and 6 in Table 1. Meanwhile, the pre-processing for the sentiment analysis libraries utilised steps 1, 2.2, 3 and 5 in Table 1 as the initial pre-processing steps.

Table 1: Pre-processing Steps

Step	Pre-process	Description
1	**Expanded contractions**	Words such as *don't*, *we'll*, and *I've* have been expanded to ensure more accurate processed text.
2.1	**Removed punctuation**	Symbols or other specials such as *!*, *]*, and *:* are removed to simplify the text and clean it up.
2.2	**Remove some punctuation**	We want to remove symbols which cannot be interpreted as emojis such as *!* and *?* while maintaining symbols that can be associated with an emoji, such as *)* and *;*.
3	**Removed stop words**	Words such as *a*, *is*, and *for* are removed since they bring little to no actual information for the models.
4	**Filtered only alphabetic characters**	Making sure each token contains alphabetic characters to further filter out unnecessary symbols that might have been missed previously.
5	**Lowercasing**	All letters are simplified to lowercase.
6	**Lemmatization**	Turn words into their base form such as turning *changing*, *changes*, and *changed* to *change*.

4 Method

This chapter presents all the steps taken to achieve the result of this report.

4.1 Pre-processing

The first step of performing the sentiment analysis and political lean prediction is pre-processing the dataset. This was done using *spaCy's* trained pipeline *en-core-web-lg* to have a large vocabulary available. Exactly what was performed is described in section 3.2.

When the initial pre-processing was done, the data was split into a training set and a test set which consisted of an 80/20 ratio. This means that approximately 10400 posts and their respective political lean label were used for training while the remaining 2600 posts were used for testing. The number of posts described refers to the initial dataset. When undersampling was utilised, the dataset became smaller to balance the dataset.

To finalise the pre-processing stage to allow the models to be trained, the data was vectorised using the *sci-kit learn* library and specifically the *TfidfVectorizer*. This was chosen over e.g. *CountVectorizer* because, with the TfidfVectorizer, we can acquire better accuracy. Also, the TF-IDF approach focuses on the frequency of words present in the corpus and provides the importance of the words analysed [13]. This will be crucial when performing sentiment analysis on words commonly paired with a political lean and their entire sentences.

4.2 Models

The *Dummy* model will be used as a baseline, while the models chosen for political lean prediction were *Multinomial Naive Bayes*, *Bernoulli Naive Bayes*, *Linear SVM* and *Random Forest* which are all supervised learning approaches. However, they differ in how they approach their training, which is described in chapter 2. Each model was retrieved from the *sci-kit learn* library. All models will utilise a set *random state* of 42 to ensure reproducibility. This number was chosen at random.

The models chosen for the sentiment analysis were the third-party libraries VADER and TextBlob which both have specifically designed capabilities for determining the sentiment of a text. Both were chosen for their popularity and to be able to be evaluated properly. For TextBlob, two sentiment analysers were available which were mentioned in Section 2.6. The top words from the Multinomial Naive Bayes and the Bernoulli Naive Bayes model were used to only find the sentiment with sentences using those words. This was done by searching through every post to find the keyword while making sure the same row was not analysed multiple times to avoid duplicate analysis. This was done by transforming the found sentences to a set which removes duplicates to later convert back to a list when performing the sentiment analysis. The sentiment was measured using the polarity score retrieved by using the third-party libraries. When categorising the sentiment of each post, a sentiment polarity between -0.1 and 0.1 was considered neutral while anything lower than -0.1 and higher than 0.1 were considered negative and positive respectively. Together with the percentage of each sentiment, the total of rows used for the percentage calculation was calculated by getting the length of the list of all sentences found with the specific keywords.

4.3 Evaluation of Classifiers

Evaluating the different models, scores such as precision, recall, F1-score and so on will be considered between the models and approaches. This scoring system is commonly used when evaluating classifiers, which is what Kalcheva et. al. used when they compared two Naive Bayes classifiers, for instance, [5]. This scoring system applies to political lean prediction and sentiment analysis. These are acquired using *sci-kit learn's metrics*, specifically *classification_report* which generates a report containing the precision, recall, f1 and other averages. Scores from three different runs will be retrieved. The initial evaluation will run without any modifications to the data or models. The second round will utilise undersampling of the majority class to investigate whether balancing the dataset might improve the scores.

4.4 Undersampling

After the initial evaluation, the liberal data will be undersampled to balance the dataset and see if a more balanced prediction precision, recall and F1-score can be achieved at the expense of accuracy for liberal posts. A boolean was used to prepare the data with the chosen undersampling technique, specifically inserting *True* to the prepare_data function [2]. The undersampling is performed by identifying the lowest amount of a political lean and then scaling down the majority class, which in this dataset is liberal.

4.5 Sentiment analysis

The Reddit posts which contained the most decisive words had to be analysed whether the sentiment of those posts was positive, neutral or negative. This was done using two different libraries to evaluate their performances respectively. The first one was VADER while the second library used was TextBlob, which were described in Section 2.5 and Section 2.6 respectively.

5 Results

This chapter presents the results for the initial models and undersampled models, as well as the results of the sentiment analysis. For all the tables presented, the header represents the following:

- Acc. => Accuracy
- Prec. => Precision
- Rec. => Recall

5.1 Initial Models

The initial models produced a result which can be observed in Table 2, Table 3 and Table 4. From the results of the initial models, we can observe that all models had a significantly increased F1-score compared to the baseline for the liberal predictions. However, this was not the case for the conservative predictions. Even though an increase was noted, it was not as significant but still noticeable. This was however not the case for the Multinomial classifier, which despite having high precision, since its recall results were poor. Random Forest came out as the best performer in F1-score and overall accuracy, with Linear SVM and Bernoulli Naive Bayes being a close second.

Table 2: Initial Models Results: Liberal

Model (Classifier)	**Prec. (%)**	**Rec. (%)**	**F1 (%)**
Dummy	0.64	0.50	0.41
Multinomial	0.71	0.98	0.82
Bernoulli	0.77	0.92	0.84
Linear SVM	0.80	0.86	0.83
Random Forest	0.79	0.90	0.84

Table 3: Initial Models Results: Conservative

Model (Classifier)	**Prec. (%)**	**Rec. (%)**	**F1 (%)**
Dummy	0.35	0.49	0.41
Multinomial	0.87	0.27	0.41
Bernoulli	0.77	0.50	0.61
Linear SVM	0.70	0.60	0.65
Random Forest	0.77	0.57	0.65

Table 4: Initial Models Results: Accuracy

Model (Classifier)	**Acc. (%)**
Dummy	0.5
Multinomial	0.73
Bernoulli	0.77
Linear SVM	0.77
Random Forest	0.78

5.2 Undersampled Models

After the initial models were evaluated, the dataset's majority class (liberal) was undersampled to balance the dataset, this resulted in what can be observed in Table 5, Table 6 and Table 7. To summarise the results, the conservative predictions had a lower precision than with the use of the unbalanced dataset. However, the overall recall for conservative predictions increased. This is especially noticeable for the Multinomial classifier which improved with more than 30%. This in turn made the F1-score across both conservative and liberals predictions more balanced after undersampling was performed. One outlier to this is the Bernoulli classifier. However, the difference is not significant. The overall accuracy decreased slightly compared to the initial results. Random Forest and Linear SVM performed well with the undersampled models as well.

Table 5: Undersampled Models Results: Liberal

Model (Classifier)	**Prec. (%)**	**Rec. (%)**	**F1 (%)**
Dummy	0.52	0.52	0.52
Multinomial	0.74	0.72	0.73
Bernoulli	0.78	0.56	0.65
Linear SVM	0.73	0.73	0.73
Random Forest	0.74	0.74	0.74

Table 6: Undersampled Models Results: Conservative

Model (Classifier)	Prec. (%)	Rec. (%)	F1 (%)
Dummy	0.52	0.52	0.52
Multinomial	0.72	0.75	0.74
Bernoulli	0.65	0.84	0.74
Linear SVM	0.73	0.73	0.73
Random Forest	0.74	0.73	0.73

Table 7: Undersampled Models Results: Accuracy

Model (Classifier)	Acc. (%)
Dummy	0.52
Multinomial	0.73
Bernoulli	0.70
Linear SVM	0.73
Random Forest	0.74

5.3 Sentiment analysis

Every post containing the top ten most decisive words for each model was included in the sentiment analysis to analyse the sentiment of posts with commonly used words for both political leans. Results for the liberal sentiments can be observed in Table 8 and Table 9 while the results for the conservative sentiments can be observed in Table 10 and Table 11. These tables present the top words and their percentage of positive, negative and neutral sentiment where the top words existed in the post. This process was done based on the top words from the Multinomial Naive Bayes classifier and the Bernoulli Naive Bayes classifier. For each word, the number of rows where the keyword was used for the sentiment analysis is represented by (#), excluding duplicates of sentences. For both classifiers, it can be observed that

Table 8: Top 10 Words for Classifying as 'Liberal' with Sentiment Analysis using Multinomial

Multinomial Liberal							
	VADER			TextBlob			
Word	Pos (%)	Neu (%)	Neg (%)	Pos (%)	Neu (%)	Neg (%)	#
Biden	35.09	27.63	37.28	30.70	58.77	10.53	228
Capitalism	52.55	15.33	32.12	39.42	56.93	3.65	137
DeSantis	15.00	40.00	45.00	15.00	60.00	25.00	20
Government	58.64	3.70	37.65	39.51	51.85	8.64	162
People	54.13	9.47	36.41	36.89	54.13	8.98	412
Putin	31.43	14.29	54.29	14.29	77.14	8.57	35
Russia	40.62	15.62	43.75	18.75	71.88	9.38	128
Say	44.79	18.93	36.28	33.75	56.47	9.78	317
Trump	25.47	27.36	47.17	24.84	60.38	14.78	318
Ukraine	33.02	27.36	39.62	25.47	66.04	8.49	106

Table 9: Top 10 Words for Classifying as 'Liberal' with Sentiment Analysis using Bernoulli

Bernoulli Liberal							
	VADER			TextBlob			
Word	Pos (%)	Neu (%)	Neg (%)	Pos (%)	Neu (%)	Neg (%)	#
Biden	35.09	27.63	37.28	30.70	58.77	10.53	228
Capitalism	52.55	15.33	32.12	39.42	56.93	3.65	137
DeSantis	15.00	40.00	45.00	15.00	60.00	25.00	20
Government	58.64	3.70	37.65	39.51	51.85	8.64	162
People	54.13	9.47	36.41	36.89	54.13	8.98	412
Putin	31.43	14.29	54.29	14.29	77.14	8.57	35
Russia	40.62	15.62	43.75	18.75	71.88	9.38	128
Say	44.79	18.93	36.28	33.75	56.47	9.78	317
Trump	25.47	27.36	47.17	24.84	60.38	14.78	318
Ukraine	33.02	27.36	39.62	25.47	66.04	8.49	106

Table 10: Top 10 Words for Classifying as 'Conservative' with Sentiment Analysis using Multinomial

Multinomial Conservative							
	VADER			TextBlob			
Word	Pos (%)	Neu (%)	Neg (%)	Pos (%)	Neu (%)	Neg (%)	#
Biden	31.54	33.41	35.05	25.23	63.08	11.68	428
Election	45.45	16.36	38.18	34.55	58.18	7.27	55
Like	65.55	2.24	32.21	38.94	54.62	6.44	357
New	43.45	20.77	35.78	50.80	41.53	7.67	313
People	51.91	7.64	40.45	39.33	50.79	9.89	445
Right	43.40	16.60	40.00	44.91	50.19	4.91	265
Say	46.75	17.21	36.04	36.69	54.55	8.77	308
Think	57.14	8.21	34.64	43.57	49.29	7.14	280
Trump	31.54	33.41	35.05	25.23	63.08	11.68	428
Woman	14.29	7.14	78.57	17.86	50.00	32.14	28

Table 11: Top 10 Words for Classifying as 'Conservative' with Sentiment Analysis using Bernoulli

Bernoulli Conservative							
	VADER			TextBlob			
Word	**Pos (%)**	**Neu (%)**	**Neg (%)**	**Pos (%)**	**Neu (%)**	**Neg (%)**	**#**
People	51.91	7.64	40.45	39.33	50.79	9.89	445
Like	65.55	2.24	32.21	38.94	54.62	6.44	357
Think	57.14	8.21	34.64	43.57	49.29	7.14	280
New	43.45	20.77	35.78	50.80	41.53	7.67	313
Know	53.85	8.97	37.18	36.75	57.69	5.56	234
Right	43.40	16.60	40.00	44.91	50.19	4.91	265
Say	46.75	17.21	36.04	36.69	54.55	8.77	308
Work	58.36	13.38	28.25	40.89	51.67	7.43	269
Trump	31.54	33.41	35.05	25.23	63.08	11.68	428
Want	60.47	8.37	31.16	43.26	51.63	5.12	215

6 Discussion

This chapter will provide a discussion of the results presented in Chapter 5. Topics such as how we can interpret the results of the classifiers and sentiment analysis will be covered. Furthermore, comparing the results to scientific papers will be conducted to evaluate where appropriate, and to see whether the results seem reasonable.

6.1 Political Lean Prediction

The overall result of the political prediction is considered successful. The initial result showed that it was achievable to get results showing that the models performed better than the baseline. This means the models are better at identifying positive cases while avoiding false positives and negatives. It was especially noticeable for the liberal results. Logically, the liberal class predictions performed better than the conservative class predictions because the liberal class was the majority class, and therefore had more training data to utilise. As observed in Section 5.1, Random Forest performed the best when using the initial dataset. Linear SVM and the Bernoulli Naive Bayes classifiers were close runner-ups, making them viable choices as well. The Multinomial Naive Bayes classifier performed noticeably better when more data were available of the specific class, which is a limitation of the original dataset. If more data had been available to train on, the results of all models would likely represent the performance capabilities of each model better. However, this does not mean that all results would improve since more data could either lead to overfitting or showing the true capabilities of each model.

For the undersampled models, it can be observed in Section 5.2 that the balance between the F1-score between liberal and conservative predictions became more balanced compared to the initial models while still performing better than the baseline. This result is considered successful because of these reasons. The decrease in liberal F1-score and the increase in conservative F1-score is logical due to the now balanced dataset. However, by using the undersampled dataset, the performance of Random Forest, Linear SVM and Bernoulli worsened. This means that out of all the predictions that were made for each model, there were fewer correct predictions. Since the overall accuracy takes both the liberal and the conservative predictions into account, we can conclude that undersampling the dataset made the model incorrectly guess a liberal political lean more times than we correctly guessed a conservative political lean, compared to the initial results of the mentioned models. However, we succeeded in balancing the overall precision, recall, F1-score and accuracy with the use of the undersampled dataset.

For both rounds of predictions, the results could be improved even further or bring even deeper analysis if we take more columns from the original dataset into account, such as what specific subreddit the post was posted at. This could lead to discussions as to whether different political leans tend to post on subreddits where people would be inclined to agree (subreddits with the same political leans as the poster) or where they can trigger or wake a discussion (subreddits with a different political lean compared to the poster). Oversampling could also be a direction to explore further.

6.2 Sentiment Analysis

The overall sentiment analysis is considered successful, as we managed to get a trend of results depending on what model and what word was considered. By taking a look at the actual titles and texts, the results that were produced can be qualified as valid. It is important to note that humans interpret the sentiment differently, especially when it comes to posts that have a more neutral sentiment.

It was observed that the top words and sentiment percentages between the Multinomial and Bernoulli Naive Bayes classifiers were identical in the majority class. This could show that sentiment analysis can be consistent for majority classes between models that are similar in their foundation.

However, discrepancies in this trend were found for the conservative class, we can see that this is not the case. This could suggest that the process of retrieving the top words of minority classes can be more complex than the majority class. Further analysis could be done to see whether minority classes have to have their top words extracted using different measures compared to the majority class.

When comparing VADER and TextBlob, the findings show that VADER tended to label the overall sentiment as positive or negative more often compared to TextBlob which heavily favoured ranking the sentiment as neutral, rather than negative. This could suggest that VADER could be a more suitable sentiment analyser if the intensity of the sentiment is more important compared to the overall sentiment. Further analysis showing intervals of the polarity scores could provide with deeper analysis of VADER. Compared to a study conducted by Thapa shows that this project utilised a mix of base polarity (boundary points are ± 0) and moderate polarity (boundary points are ± 0.25) since our boundary points were ± 0.1, as described in Section 4.2. The results from Thapa's study show that Reddit posts tended to be more neutral, rather than positive or negative, regardless of the polarity's boundary points. However, this trend was even more clear when the boundary points increased [14]. Since this project suggests that VADER were more likely to categorise the sentiment as either positive or negative, rather than neutral, suggests that the context of the posts could have a large impact on the predicted sentiment.

Frequent words as presented in Section 5.3, were words such as *People*, *Trump*, *Biden* and *Say*. For the names Trump and Biden, the general sentiment was mixed according to VADER while they were mostly neutral according to TextBlob. This suggests that the political lean of the posters was likely mixed by interpreting the TextBlob results, while posters being positive or negative were more clearly negative than positive. This is backed up by the fact that VADER tended to show the intensity of the sentiment while TextBlob showed the general sentiment. It also checks out by acknowledging the fact that people with negative sentiments tend to express themselves more clearly compared to people with positive sentiments. For the words *People* and *Say*, the case was similar compared to Trump and Biden with a trend showing that the words were used more commonly in posts with a positive sentiment. However, this could be because these types of words are more general and do not carry emotion in them like Trump and Biden do, making them applicable in both positive and negative posts.

An important limitation of this study is that this regards single words only and not bigrams or trigrams. Further analysis could allow for the evaluation of top bigrams and trigrams.

7 Conclusion

In conclusion, we can say that every model evaluated for political lean prediction performed better than the baseline based on the F1-score results. However, in some cases, the model performed worse than the baseline. One example of this is the recall of the Multinomial Naive Bayes when using an unbalanced dataset. We can also conclude that overall, Random Forest and Linear SVM performed the best across the use of the unbalanced and balanced dataset.

As for the sentiment analysis, we found that the emotional tone of the two political leans had similar trends, where VADER tended to better illustrate the intensity of the sentiment compared to TextBlob, which instead produced a result which represented the general sentiment of a post. It can also be concluded that the context of the post could have a significant impact, even if different boundary points are utilised. Finally, we can conclude that analysing the top words for a majority and minority class can differ from each other, with a suggestion that further research into this matter could debunk this question. Overall, the goals of the project were fulfilled, with multiple paths of further research available with the given results.

References

[1] Neel Gajare. *Liberals vs Conservatives on Reddit [13000 posts]*. 2023. URL: https://www.kaggle.com/datasets/neelgajare/liberals-vs-conservatives-on-reddit-13000-posts (visited on 12/16/2023).

[2] Gabriel Hult. *tdde16_project*. 2023. URL: https://github.com/gabrielhult/tdde16_project (visited on 01/04/2023).

[3] C. Hutto and Eric Gilbert. "VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text". In: *Proceedings of the International AAAI Conference on Web and Social Media* 8.1 (May 2014), pp. 216–225. DOI: 10.1609/icwsm.v8i1.

14550. URL: https://ojs.aaai.org/index.php/ICWSM/article/view/14550.

[4] IBM. *What is a Decision Tree?* 2023. URL: https://www.ibm.com/topics/decision-trees#:~:text=A%20decision%20tree%20is%20a,internal%20nodes%20and%20leaf%20nodes. (visited on 01/11/2024).

[5] Neli Kalcheva, Ginka Marinova, and Maya Todorova. "Comparative Analysis of the Bernoulli and Multinomial Naive Bayes Classifiers for Text Classification in Machine Learning". In: *2023 International Conference Automatics and Informatics (ICAI)*. 2023, pp. 28–31. DOI: 10.1109/ICAI58806.2023.10339077.

[6] P. Karthika, R. Murugeswari, and R. Manoranjithem. "Sentiment Analysis of Social Media Network Using Random Forest Algorithm". In: *2019 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS)*. 2019, pp. 1–5. DOI: 10.1109/INCOS45849.2019.8951367.

[7] Kainat Khan and Sachin Yadav. "Sentiment analysis on covid-19 vaccine using Twitter data: A NLP approach". In: *2021 IEEE 9th Region 10 Humanitarian Technology Conference (R10-HTC)*. 2021, pp. 01–06. DOI: 10.1109/R10-HTC53172.2021.9641515.

[8] S. B. Kotsiantis. "Decision Trees: A Recent Overview". In: *Artificial Intelligence Review* 39.4 (Apr. 2013), pp. 261–283. ISSN: 1573-7462. DOI: 10.1007/s10462-011-9272-4. URL: https://doi.org/10.1007/s10462-011-9272-4.

[9] Steven Loria. *TextBlob: Simplified Text Processing*. 2023. URL: https://textblob.readthedocs.io/en/dev/index.html (visited on 12/28/2023).

[10] Kevin P Murphy et al. "Naive bayes classifiers". In: *University of British Columbia* 18.60 (2006), pp. 1–8.

[11] RedditInc. *Dive Into Anything*. 2023. URL: https://www.redditinc.com/ (visited on 12/27/2023).

[12] scikit. *RandomForestClassifier*. 2023. URL: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (visited on 12/26/2023).

[13] Kristien Margi Suryaningrum. "Comparison of the TF-IDF method with the count vectorizer to classify hate speech". In: *Engineering, MAthematics and Computer Science (EMACS) Journal* 5.2 (2023), pp. 79–83. DOI: 10.21512/emacsjournal.v5i2.9978.

[14] Bipun Thapa. *Sentiment Analysis of Cybersecurity Content on Twitter and Reddit*. 2022. arXiv: 2204.12267 [cs.CL].

[15] Yujun Yang, Jianping Li, and Yimei Yang. "The research of the fast SVM classifier method". In: *2015 12th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*. 2015, pp. 121–124. DOI: 10.1109/ICCWAMTIP.2015.7493959.

[16] Sheresh Zahoor and Rajesh Rohilla. "Twitter Sentiment Analysis Using Machine Learning Algorithms: A Case Study". In: *2020 International Conference on Advances in Computing, Communication Materials (ICACCM)*. 2020, pp. 194–199. DOI: 10.1109/ICACCM50413.2020.9213011.