P-stance Analysis

Drobnitchi Daniel, Suto Robert-Lucian
411

January 2023

1 Introduction

Political stance is the application of Natural Language Processing techniques to analyze and categorize text data based on political ideologies or beliefs. It involves using algorithms to identify patterns and features in text data that can predict the author's political beliefs or views on political issues. This can be used to categorize text data into political groups such as left-leaning, right-leaning, or neutral. These techniques are used in various fields such as politics, marketing, and social media analysis.

2 Dataset

For this example, we've decided to use the P-Stance dataset of the 2020 Presidential Election in USA. The dataset is organized based on the candidates : Donald Trump, Bernie Sanders, Joe Biden. Each president has it's own training and testing dataset, each containing 6500 and 800 tweets each. Each row of the dataset contains 3 columns:

- Tweet : A tweet concerning a user's view on the certain candidate.
- Target : The target, being one of the candidates.
- Stance : Represented by the tweet's opinion: Either in Favor or Against the certain candidate.

The scope of the model in this case is to conclude if a tweet is either in favor or against, no matter the candidate it is used tweeted at and without using that as a feature. This adds complexity to the way features are separated in the training part of the project, as it lacks crucial information in which it can build on.

Examples for the dataset:

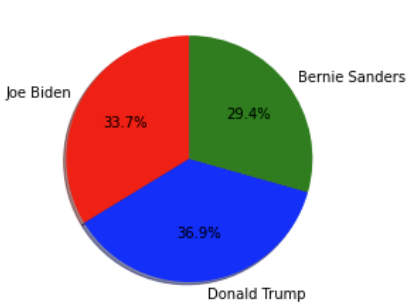
1. Against - Donald Trump: Tell us something we didnt know. Bribery Corruption Obstruction. Thats his MO. #Trump
2. Pro - Donald Trump: Attorney General Bill Barr is a rotten piece of .#MuellerReport #BillBarr #Trump
3. Against - Joe Biden : #Biden can not command respect in this little room much less our nation.
4. Pro - Joe Biden : Joe gives us solutions. Joe gives us a clear path forward while we navigate this global crisis. The next Commander-In-Chief, everyone #Twill
5. Against - Bernie Sanders : Well, a sad day for all my communist friends. I guess they are gonna have to move to Cuba or wait another 4-8 years #BernieSanders Feel the burn!!!
6. Pro - Bernie Sanders I can legally change my address to MI and just did so I can volunteer vote for #BernieSanders in MI.

It is important to take into account the relative bias a social media app has towards some of the candidates, as, for example, there were more tweets against Donald Trump compared to Bernie Sanders or Joe Biden. Either way, the following figures will explain this better.

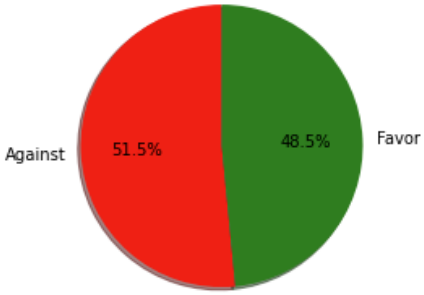
3 Exploratory data analysis

We attached some graphs and figures to accentuate the way the dataset is split.

Adding on top of this, we used some wordclouds for the pre-processed occurences of words.

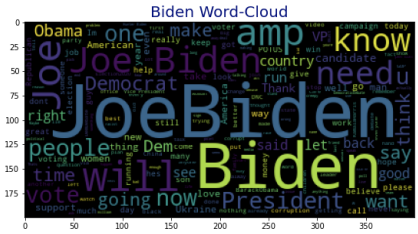


(a) Candidates

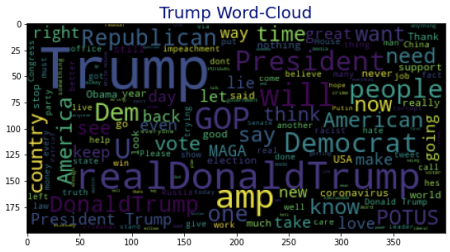


(b) Stances

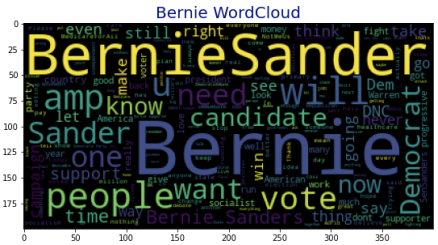
Figure 1: Graphs for distribution of data



(a) Biden



(b) Trump



(c) Bernie

Figure 2: Graphs for distribution of data

4 Processing the data

For the preprocess function we have used the following procedures:

1. Lowered all the characters.
2. Split the tweets into separate words.
3. Removed mentions from the words.
4. Stemmed each word using SnowBallStemer.
5. Unidecoded the text.
6. Removed alphanumerical characters.
7. Removed whitespaces.

To gather up the processing data, we have used Term Frequency - Inverse Document Frequency vectorizer.

5 Training the models.

We have used five different models on this dataset, along with some hyperparameter tuning using gridsearch:

- Support Vector Machine
- Logistic Regression
- Knn Neighbors
- Multinomial Naive Bayes
- Decision Tree Classifier

5.1 Support Vector Machine

Support Vector Machine (SVM) is a type of supervised machine learning algorithm used for classification and regression analysis. It finds the hyperplane in high dimensional space that best separates the data into classes and tries to maximize the margin between the two classes.

Kernel	C	Gamma	Training Accuracy
linear	1	0.001	0.575
linear	2	0.001	0.569
linear	3	0.001	0.566
linear	1	0.01	0.553
linear	2	0.01	0.532
linear	3	0.01	0.527

In the end, the training accuracy was far worse than the test one. The chosen final fit was using C=1, Gamma = 0.001 and Linear Kernel. It achieved an accuracy of 55.9% on the training dataset and 69% on the test one.

5.2 Logistic Regression

Logistic Regression is a statistical method used for binary classification problems in NLP. It estimates the probability of an event occurring based on the input features, and outputs a binary decision (e.g. positive/negative sentiment). Logistic Regression can be trained using annotated text data and make predictions on new unseen text data.

Here is the result on the grid search algorithm.

C	Solver	Score	Time
1	lbfgs	0.590	1.3s
1	newton-cg	0.590	0.4s
1	saga	0.590	0.3s
2	lbfgs	0.592	1.8s
2	newton-cg	0.592	0.3s
2	saga	0.592	0.2s
3	lbfgs	0.587	1.0s
3	newton-cg	0.587	0.3s
3	saga	0.588	0.2s

In the end, the final accuracy on the training was 59% and the final accuracy on the test one was 68.7%. With C=1, and the solver being newton-cg.

5.3 Knn Neighbors

KNN (K-Nearest Neighbors) is a non-parametric and lazy learning algorithm used for classification and regression tasks. It's based on the idea of assigning a class label to a new data point by finding its K nearest neighbors in a training dataset and taking a majority vote among them.

Algorithm	Leaf Size	k-Neighbors	Score
Auto	1	5	0.561
Auto	1	10	0.567
Auto	1	20	0.578
Auto	1	50	0.583
Auto	2	5	0.561
Auto	2	10	0.567
Auto	2	20	0.571
Auto	2	50	0.587

In the end, the final accuracy on the training was 58.3% and the final accuracy on the test one was 68.7%. leaf size = 1, and k-neighbors being 20.

5.4 Decision Tree

A decision tree classifier is an algorithm used for classification tasks that splits data into smaller subsets based on the most significant features or attributes, creating a tree-like structure. Each internal node in the tree represents a test on an attribute, each branch represents the outcome of the test, and each leaf node assigns a class label to a data sample.
Bellow are some of the results, randomly selected from a gridsearch.

Criterion	Max Depth	Min Samples Split	Score
gini	None	2	0.534
gini	None	2	0.627
gini	None	2	0.584
gini	None	5	0.537
gini	None	5	0.581
gini	5	2	0.557
gini	5	5	0.504
gini	5	5	0.531
gini	5	5	0.557
gini	5	5	0.560

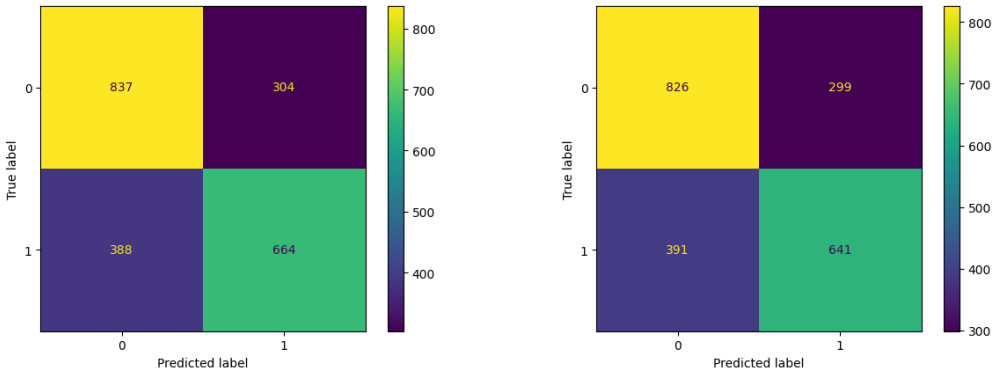
In the end, the final accuracy on the training was 57.9% and the final accuracy on the test one was 61.7%. criterion = entropy, max_depth = None and min_samples_split being 5.

5.5 Multinomial NB

Multinomial Naive Bayes (NB) is a simple and fast algorithm for text classification problems that assumes independence between features. It is a popular choice for text classification as it can handle discrete data such as word counts, and it is also scalable to large datasets. It estimates the probability of each class by counting the frequency of words in each class, and then classifies a new document based on the maximum probability.
For this algorithm we obtained an accuracy of 68.7%.

6 Conclusion

In conclusion, the results of the experiment showed that the best algorithm for predicting tweet sentiment towards political candidates was Support Vector Machines (SVM), with an accuracy of 69% on test data. The evaluation metrics indicate that the algorithms have a tendency to predict tweets as being against a candidate, rather than in favor of them, as evidenced by the high precision and low recall for the "FAVOR" class. To address this imbalance, it may be beneficial to increase the weight given to the "FAVOR" class in future iterations.



(a) Validation data

(b) Test data

Figure 3: Confusion Matrices