IJSRA

International Journal of Science and Research Archive

eISSN: 2582-8185

Cross Ref DOI: 10.30574/ijsra

Journal homepage: https://ijsra.net/

Int. J. Sci. Res. Arch.
IJSRA
International Journal of Science and Research Archive
Research Journal Archive, INDIA

(RESEARCH ARTICLE)

Social media (YouTube) political sentiment multi-label analysis

Mayowa Timothy Adesina * and Luke Howe

Data Analytics Department, College of Business, Kansas State University, KS, USA.

International Journal of Science and Research Archive, 2024, 12(02), 2063–2071

Publication history: Received on 05 July 2024; revised on 12 August 2024; accepted on 14 August 2024

Article DOI: https://doi.org/10.30574/ijsra.2024.12.2.1429

Abstract

In an era where presidential elections and political discourse are increasingly digital, understanding public sentiment through online comments becomes crucial. This project explores the relationship between YouTube comments and political preferences, challenging the null hypothesis that asserts no connection between contextless comments and candidate preference. Utilizing various natural language processing (NLP) tools, including Bing-Liu, Vader Sentiments, and Large Language Models (LLMs) like ChatGPT-4, we delve into multi-label sentiment analysis for political figures.

Our methodology encompassed a rigorous data collection process from YouTube, leveraging custom scrapers and the computational power of KSU's BEOCAT servers. We navigated through challenges like content limitations and the need for comment sanitization to comply with community guidelines. The study tested different models, including logistic regression and Graph Convolutional Networks (GCNs), against a baseline of Max Label/Zero R classification.

Results showed varying degrees of success, with individual label accuracies ranging from moderate to high. However, the overall accuracy of our final model using GCNs stood at 39%, indicating the complexity and difficulty of multi-label classification in political sentiment analysis yet also the success of graph convolutional networks at identifying complex contextless sentiment. This project not only sheds light on the potential of NLP in political analytics but also opens up avenues for future work in real-time sentiment analysis during political events, albeit with ethical considerations.

This research contributes to the growing field of NLP implementation in public opinion analysis, with implications extending beyond politics into other consumer-centric industries.

Keywords: Machine Learning; Artificial Intelligence; Social Media; Text Analysis; Natural Language Processing (NLP)

1. Introduction

This project does not exist to endorse or disparage any person or organization.

1.1. Purpose

Presidential elections and political discussions are becoming more commonplace and, in the digital age, more public. This can be valuable information to anyone involved in the process. Campaign managers can use this information to determine overall sentiment in different demographics based on who is being discussed, where, and how. Utility for these results are as varied as the customer's imagination but include:

- Checking to see which topics are trending positively or negatively for a candidate.
- Searching overall online rhetoric surrounding a candidate.
- Querying hot topics to see if candidates are being discussed at all.

* Corresponding author: Mayowa Timothy Adesina

- Querying other candidates to see where their strengths and weaknesses amongst a target demographic lies.

With this in mind, a candidate can quickly identify areas to emphasize, avoid, target, or even to study up on. For example, our data suggests that Donald Trump is very popular on economy related topics. His campaign team could focus future talking points and rhetoric on economic successes and attack opponents on their positions.

1.2. Null Hypothesis

Candidate support cannot be determined based on YouTube comments alone. Additional Context such as type and title of video, user history, and video context are required.

2. Related Work

Natural language processing is a well-studied field with other researchers who have built a variety of tools for different purposes. The first example of this are simple systems based on libraries such as Bing-Liu or Vader Sentiments. These strategies extend into much more commonly thought of and complex example of this are large language models (LLMs) such as Google's BARD.

2.1. Background

2.1.1. BING-LIU

Bing-Liu text analysis is possibly the simplest form of text analysis. So simple that one might not even consider it to be text analysis. Bing-Liu is little more than a string search program with a pre-loaded dictionary. That is, Bing-Liu searches for words that are in a pre-loaded, though customizable, dictionary and then tally the ratio of "good words" versus "bad words". The most obvious limitation here is the total loss of context. Context isn't even considered which can cause extreme confusion when it comes to any statements that aren't straightforward. For example, in our data set "This country is going to hell in a handbasket! We need (candidate)!" would be classified as a simple "negative" because of the harsh language. Bing-Liu might be useful for a quick and computationally cheap look at very simple statements such as online restaurant reviews. From previous research, we've seen that restaurant reviews are typically one to two sentences that convey strong emotion in plain language. Statements like "Service was terrible" or "I loved the pasta!" are very easy to identify with Bing-Liu. However, without a strong operating knowledge of the topic, a researcher could fail to clarify a great deal of comments without the ability to capture misspellings in some way. Furthermore, there is no Bing-Liu method of attributing a comment to the person that made it. That is, even if we could appropriately determine whether a comment had a "positive" or "negative" sentiment that doesn't indicate who it was directed towards or if it actually expressing support but out of anger such as in the aforementioned sample comment. Another situation that Bing-Liu can't identify is sarcasm or doublespeak, "yeah, right" would be a positive Bing-Liu comment. Finally, comments that aren't strongly positive or negative but have a polarity for or against a politician's standpoints will be rated as neutral. All of these issues mean that Bing-Liu isn't even viable as a base model to compete against.

2.1.2. Vader sentiments

Vader Sentiments is also a basic analysis tool that is computationally and conceptually simple. However, unlike Bing-Liu, it is a pre-trained model that excels at determining positive, neutral, or negative sentiments and built to do so on social media comments specifically. Vader is also able to handle emojis which preserves a great deal of context that's otherwise lost in text only analysis. Vader is case sensitive as well, this allows us to differentiate between "do something about it" and "DO SOMETHING ABOUT IT". In keeping with this more context aware approach, VADER can VADER uses ngram windows to catch additional context that Bing-Liu can't. Bing-Liu would rate "not bad" as a negative statement but Vader is better able to recognize and react to that context. Though this sounds like exactly the solution we were looking for, Vader actually suffers from a lot of the same issues as Bing-Liu. Using this method, combined with named entity recognition and extraction to identify who's being mentioned, we can create a zipped column of all entities named in a comment and the overall sentiment of the comment. Let us revisit our comment from earlier "This country is going to hell in a handbasket! We need (candidate)!" In this comment Vader and the should identify "negative" and "(candidate)" however, even though it clearly expresses support for that candidate. This means Vader is also in capable of rating a comment as pro- one candidate and anti-a different one. Clearly not the ideal solution. Solving this multi-label classification problem using Vader sentiments merged with topic analysis is able to meet roughly a 15-%17% accuracy over 10 independent trials.

2.1.3. MAX LABEL/ZERO R

Another, less scientific solution could be to simply take the most common label in the dataset, associate that label with 100% of entries, and then hope for the best. Therefore, we've used this method as one of our baselines to outperform as an indication of a valuable model. Often, max label is used to judge the efficacy of an imbalanced binary classification model in systems like fraud identification, it can be very difficult to defeat a max label test. However, the max label in our dataset is representative of a plurality rather than a majority. As a result, our max label is "pro-trump" which represents 25% of the data. Since there is no model associated with this method of labeling comments, that means max label is the cheapest and easiest model to implement and with that level of accuracy, also the most accurate model.

2.1.4. Large Language Models

The gold standard in modern natural language processing is the large language model. These systems have shown great proficiency in many areas from conversational ability to cooking advice, and limited coding ability. These systems have been built trained and optimized using billions of parameters to best understand and even respond to human speech therefore they are an obvious choice for our political multi-label classification problem. Our group had access to ChatGPT 4 for testing and experimentation and decided to use this model to attempt to classify comments. One limitation of ChatGPT is the amount of data that can be loaded in at any given time for natural language processing towards the end of the project OpenAI released a data analysis plugin for their most popular and powerful model but that plugin does not use the same natural language processing techniques and power that their interactive model does and is not as effective in fact that model uses the Vader sentiments method we described earlier to classify text rather than an actual generative transformer. Therefore, our accuracy metric when using ChatGPT four and other large language models are limited by the amount of data we can actually input into the model at any given time. To generate a useful comparison of accuracy we selected 100 random lines of comments from our training data set to input into the large language model and test its success. The majority of prompts are therefore then simply the YouTube comments however the original preparatory prompt which explains to the large language model how to

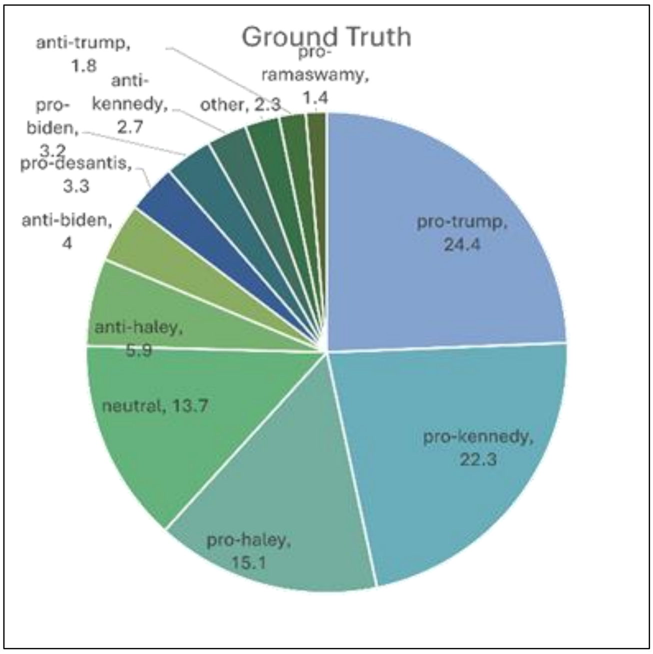

Figure 1 Ground truth labeled comments. "Pro" comments in blue, "anti" comments in red, and neutral highlighted in gray

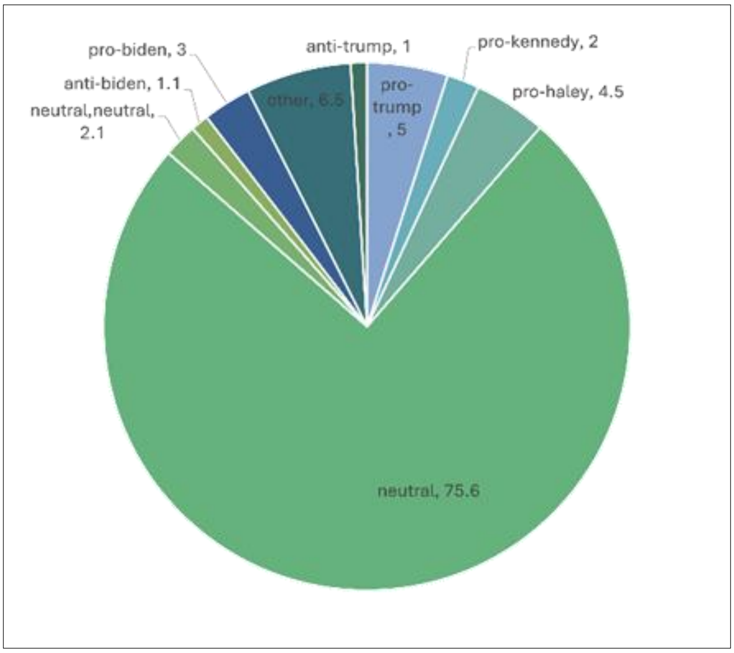

Figure 2 ChatGPT4 labeled comments. “Pro” comments in blue, “anti” comments in red, and neutral highlighted in gray

proceed with the following tests is listed below;

Prompt: “I'm going to give you a set of comments from YouTube, and I would like you to classify them using any combination of the following labels to determine the prevailing sentiment or sentiments in those comments thank you.

"pro-biden","anti-biden","pro-trump","anti-trump",”pro-christie","anti-christie", "pro-kennedy","anti-kennedy", "pro-ramaswamy","anti-ramaswamy","pro-desantis","anti-desantis","pro-haley","anti-haley",”pro-scott","anti-scott","pro-pence","anti-pence"”

The first hurdle we met in this approach was that the model typically did not like to discuss certain real-life people. I believe that colored its ability to rate certain comments as “pro” or “anti” well known political figures. The second issue we encountered was also a content violation issue, many of the YouTube comments we harvested contain shocking graphic or otherwise disagreeable and unpalatable information. Specifically, one comment that was selected at random to be processed by the large language model referenced allegations of sexual abuse towards certain people. Including that information as a comment to rate immediately disabled any chat sessions the model was using to rate comments. As a result, not all comments are truly randomly selected and some had to be cherry picked to follow community guidelines. These two situations alone are obvious reasons why a system like ChatGPT, before discussing any intrinsic biases that could be introduced by developers, are unlikely to be acceptable solutions to our political sentiment classification problem. However, large language models do excel at classifying comments for which there is little label support for in the comments. That is, some politicians such as Chris Christie are highly underrepresented in the data set and the large language models excelled at being able to identify that Chris was a person's name and whether or not that person supported him. Other models to be discussed later in the methodology section were more accurate in situations in which a person was not explicitly mentioned in a comment but only in situations where there was enough labeled training data for those models to quote get an idea of quote the speech patterns and major topics that are concerns of certain politician’s supporters or detractors. Discuss in more detail in the methodology section the strengths that other models had that large language models did not have when it came to speech pattern recognition. Comments that would otherwise need context from outside the comment itself were better classified by other models. Examples might be “USA USA USA USA USA!!!!” those comments were more common with some politicians than other politicians we found when training the models that nearly half of candidates could be ruled out as the subject of a comment based on the level of patriotism in that comment and those models were also able to identify “pro” or “anti” sentiment much more easily using that heuristic as well. Large language model accuracy could potentially be increased by further cleaning the data before presenting it to the model in that way the model might be able to better handle nicknames or pejoratives that are commonly used for certain candidates. Again, to be discussed later our final model was quite proficient at identifying that any comment mentioning the name “Brandon” was going to be an “anti-biden” comment but large language models instead would classify them as a “pro-brandon” comment. Finally, ChatGPT4’s natural language processing accuracy was 20% on our data set (ChatGPT4, 2023).

2.2. Related Work

2.2.1. Natural language processing implementation

Natural language processing is one of the most exciting nascent fields in the realm of artificial intelligence. As alluded to in the introduction, there are many uses for this technology in the realm of politics which is really the realm of public opinion. Public opinion can be extrapolated into any field of industry or the sciences that deal with customers, clients, patients, or other consumers of a service. The work done by OpenAI over the course of the last year proves the eagerness of the market to adopt these technologies into every industry. This year, 2023, the Screen Actors Guild (SAG) held a strike for over six months out of fear of the rapid progress of generative AI (Pulliam-Moore, 2023). The potentiality of totally AI generated actors was the forefront of media coverage but equally distressing to the Hollywood writers is the prospect that their creative input could be devalued by quality and rapid idea generation by AI. To produce an artificial intelligence that is capable of replacing human ingenuity, we would first have to train it on what quality human ingenuity looks like. Models such as the one we have developed here could be used to rapidly create training data for those AI from sparsely labeled datasets. Further refinement of models and the datasets they’re built upon would be necessary, but this project demonstrates a proof-of-concept for an entire pipeline of data gathering, cleaning, processing and predicting complex belief system expression or public opinion.

3. Methodology

3.1. Introduction

Multiple models and parameters for those models were considered for this project but due to the nature of the class, and the successes seen with neural networks (NN), emphasis was placed on selecting an NN.

3.2. Data Collection

3.2.1. Project development

Candidates were chosen using polling data from www.fivethirtyeight.com, an ABC News polling website, with over 1% support1. (ABC News, 2023) Topics for validation dataset were determined by choosing politically divisive topics in the United States such as border security, climate change, the economy, or school choice. All videos were curated to be less than 1 year old to maintian relevancy to the 2024 U.S. Presidential election cycle.

3.2.2. Scraping

No premade solutions were used during this project to include the data collection step. Data was scraped by a custom-made YouTube comment scraping code (provided) for all politicians and topics. YouTube provides a robust API for collecting comments and comment metadata but charges a premium for that convenience. As a result, we developed a custom web scraper using Selenium and powered by BEOCAT to gather the training data and requisite “gigabyte level” validation data. Each politician had their entire last year of self-produced videos scraped for all comments. Most politicians had open comments on their videos but some, such as Tim Scott, Chris Christie and Mike Pence had locked many or all of their comment sections. This alone is indicative of a politician’s success as of those three, only Chris Christie is still participating in the election. After collecting the politician made videos, they were aggregated into one document and hand labeled to provide seed data to the project as LLMs were unable to provide usable labeling as described in the background of this project.

Due to the time intensive process of scraping YouTube, BEOCAT was instrumental to the project. It’s estimated that we required roughly 1600 hours of compute (operated in parallel) to collect the data.

3.2.3. Data preprocessing

Cleaning the data was, as anticipated, a very time-consuming process during this project. The comments had to be encapsulated in quotation marks to escape CSV delimiter issues caused by commas in the comments. Some null comments, non-English comments, and certain emojis had be removed as well. Most emojis were kept since many comments were contextually dependent on the emoji. The comments were tokenized to remove overly common words that might poison the algorithm. These tokenized phrases had to also be vectorized using Word2Vec to maintain a context window and in later steps, topics were extracted from the comments and appended to the vectors. Each row was pre-staged with a dummy variable for each stance on a politician such as “pro-trump” and “anti-trump.” Results could then be any combination of eighteen different dummy variables with the 19th (neutral) being omitted. There are some issues inherent to each variable being analyzed independently which will be discussed later in the Results section. The processed files were saved as Parquet files to assist with managing the size of data being processed.

3.2.4. Model construction

Logistic regression

Multi-label classification is a difficult tax that is best managed by multiple binary classification tasks. Pursuant to this, the first model evaluated was a logistic regression with 5-fold validation, parameter grid tuning, and five train/test splits tested on the data. During this process we had to use under and oversampling to generate a usable dataset due to a few politicians having few comments mentioning them and Chris Christie having not a single pleasant thing said about him across over 11 million comments. Using this method, we were only able to achieve roughly 15% accuracy which is comparable to a Vader sentiments analysis but less accurate than our max label baseline. Despite many different parameters evaluated using grid method, logistic regression was not an acceptable or accurate method.

Graph Convolutional Networks (GCN)

Since one of the group objectives was to leverage NN, a GCN model was our next choice. To construct the GCN we used tensors constructed from the vectorized words and topic selection from the comments as well. Then the topics were used to calculate cosine similarity between each comment. Fifty topics is the optimum number of topics discovered after multiple tuning iterations. Using this transformed dataset, we constructed our model using an ADAM optimizer learning rate of 0.001, a train/test split of 30/70, and 500 epochs. Hyperparameter tuning was conducted using a bracketing method assisted by BEOCAT which allowed us to run more than twenty different combinations of the code simultaneously to find the optimum set of model parameters.

3.2.5. Evaluation metrics

Individual accuracy, precision, recall, and F1 score are less effective in measuring success than in binary classification problems. In this case, the only way to calculate precision, recall, or F1score is to calculate them for each politician and sentiment separately. That is, separate calculation for pro-biden, from anti-biden, from pro-ramaswamy, from anti-ramaswamy. The dataset is obviously very sparse as a result of no topics referring to every single politician. Furthermore, any comment can hold any combination of sentiments regarding politicians. Commonly, President Biden and President Trump were used as a measuring standard in comments referring to other candidates. Therefore, the primary evaluation metric used is overall accuracy. A comment is classified as correctly identified only if every sentiment that is associated with a comment is correctly identified and no extemporaneous sentiments are associated with the comment. That is a comment that should be pro-trump, anti-biden would be marked incorrect if the prediction referenced anyone else, referenced anti-trump, pro-biden, or only one correct sentiment. To be considered as correct, it must be classified as exactly and completely correct. With this in mind, our accuracy metric is quite stringent. Our final model produced up to 39% accuracy over 5062 comments and had greatest success relative to the number of available comments for each classification. That is, greater accuracy came from greater numbers of available training data with that label or label set. As we addressed with our LLM tests, the LLMs were superior at identifying low label availability comments but frequently failed to identify ambiguous high label availability comments that GCNs built with context could excel at.

4. Experiment Design and Results

4.1. Experiment design

This experiment was designed with the knowledge in mind that natural language processing is a challenging task for any model, including the human mind. Our data was labeled by three different people and quality controlled by Luke Howe. During this process, all three people had different opinions on how to classify any given comment. That means that two different humans hand classifying this data would likely agree on half or less of the “ground truth” labelling with reasonable justification for each stance. Therefore, from the outset of this project, our goal has been to disprove the null hypothesis, thereby establishing that there is a provable link between the vectorized words in a comment and a reasonable human classified conclusion for that comment. Our goal, as stated, has been to do this by outperforming our max label classification which means that regardless of any potential human bias in label classification, our model can conform to the pattern established in the training set to produce matching label as if the same person/s were to hand label all comments.

5. Results

5.1. Individual Label Accuracy

- Accuracy of the model on the pro-biden: 0.9684
- Accuracy of the model on the anti-biden: 0.9647
- Accuracy of the model on the pro-trump: 0.8874
- Accuracy of the model on the anti-trump: 0.9814
- Accuracy of the model on the pro-christie: 1.0000
- Accuracy of the model on the anti-christie: 0.9975
- Accuracy of the model on the pro-kennedy: 0.7765
- Accuracy of the model on the anti-kennedy: 0.9726
- Accuracy of the model on the pro-ramaswamy: 0.9845
- Accuracy of the model on the anti-ramaswamy: 0.9980
- Accuracy of the model on the pro-desantis: 0.9628
- Accuracy of the model on the anti-desantis: 0.9887
- Accuracy of the model on the pro-haley: 0.8538
- Accuracy of the model on the anti-haley: 0.9396
- Accuracy of the model on the pro-scott: 0.9994
- Accuracy of the model on the anti-scott: 0.9992
- Accuracy of the model on the pro-pence: 0.9994
- Accuracy of the model on the anti-pence: 0.9977

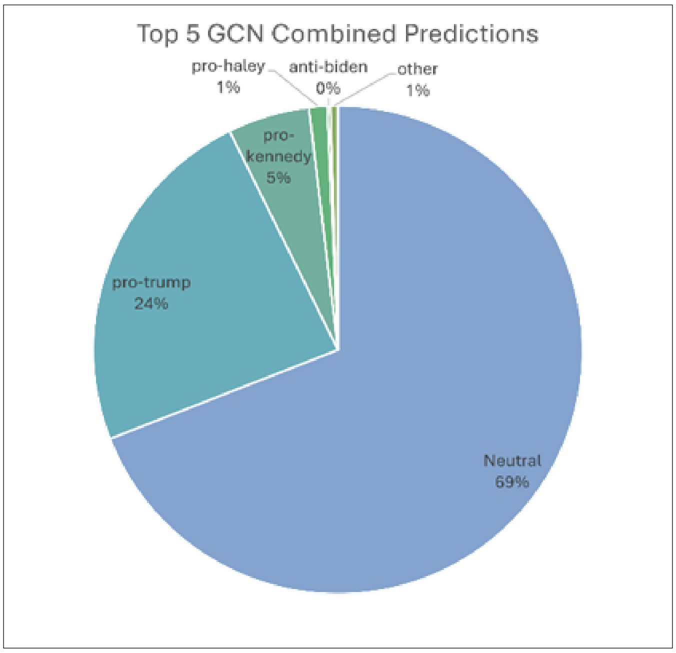

Figure 3 Top 5 GCN Prediction

Here we can see that our average ROC is hindered greatly by lower quality data labels. The eighteen different labels have widely varying quality of input data and label instances. These lower quality labels such as those related to Chris Christie or Kennedy affected the overall precision negatively. Focusing more on labels with more and higher quality data returns greater precision scores.

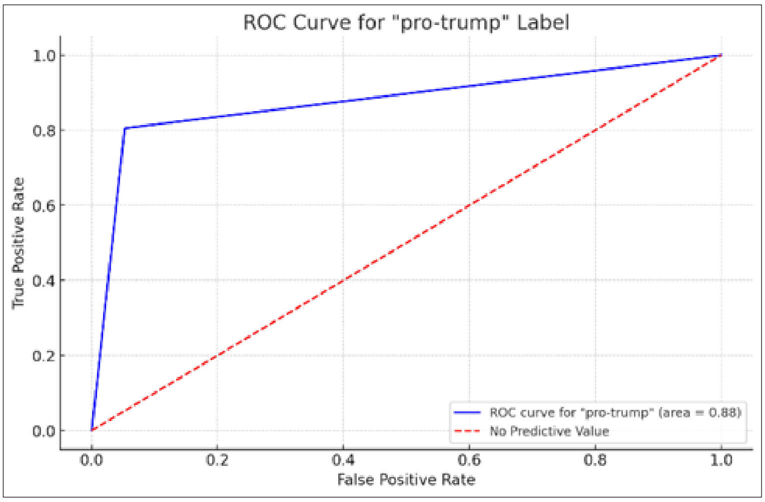

Figure 4 Pro-Trump Receiver Operating Characteristic Curve

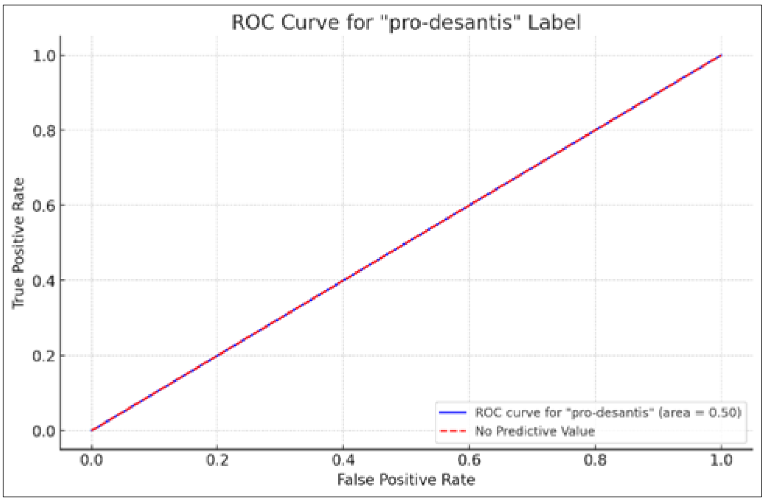

Figure 5 Pro-Desantis Receiver Operating Characteristic Curve

6. Conclusion

This project is more complex than entirely necessary because it shows the labeling for the entire race for all parties rather than a single candidate. A much simpler iteration of this project would identify comments for a single politician, potentially even using a single model which determines whether a comment is pro or anti and then classifying uncertain comments as neutral. This approach would be much more sensitive to any single candidate and likely reduce neutral findings. Our greatest results line up with our most common labels, other candidates could be reduced from the running and therefore increase accuracy if we didn't consider politicians with a popularity so low that we couldn't find comments even referring to them. That is primarily candidates such as Chris Christie. That is a problem of his own making since he locked his videos. Though likely due to a preponderance of negative feedback, we would likely still be able to find enough positive feedback to build a model capable of finding positive comments.

6.1. Future work

From the results of this study, we can see that greatest accuracy comes from a GCN model that is preloaded with plenty of data. Ideally for this task, I'd like at least 1000 comments for any given label. I'd also like to see this adapted into a true live environment during a debate so that participants and viewers can see the live sentiment shift during a speech. This idea however brings ethical considerations with it as well since it's likely that seeing mass opinion would likely have significant influence on participants. Therefore, public opinion could become quite vulnerable to model loss or even bad actors

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] ABC News. (2023, September 10). *Latest Polls*. Retrieved from fivethirtyeight.com: https://projects.fivethirtyeight.com/

[2] *ChatGPT4*. (2023). Retrieved from OpenAI: https://chat.openai.com

[3] Pulliam-Moore, C. (2023, November 18). *SAG-AFTRA's New Contract Hinges on Studios Acting Responsibly With AI*. Retrieved from The Verge: https://www.theverge.com/2023/11/18/23962349/sag-aftra-tentative-agreement-generative-artificial-intelligence-vote