Analysis of Sentiment in YouTube Comments for 2024 Political Campaigns for Governor of Puerto Rico

Carlos A. Oriol Rivera
Graduate Student Program
Advisor: Dr. Jeffrey Duffany
Department of Computer Engineering
Polytechnic University of Puerto Rico

Abstract — *The analysis of public sentiment is an invaluable tool in political campaigns, enabling candidates and their teams to assess voter attitudes and adjust campaign strategies accordingly. This project focuses on collecting and analyzing video comments on YouTube from the months leading up to an election to determine the public's sentiment toward different candidates for governor in Puerto Rico. To make this sentiment analysis, the Text Blob Python library was used due to its ease of use and robust functionality. The project incorporates both client-side and server-side components, integrating modern web development frameworks and libraries to provide good user experience. This investigation outlines the data collection methodology, system architecture, analysis process, discussion of results, and coding logic used to achieve the analysis.*

Key Terms — *Political Campaigns, Sentiment Analysis, TextBlob Python Library*.

INTRODUCTION

In today's political landscape, understanding public sentiment has become essential for creating effective campaign strategies. Social media platforms, such as YouTube, play a very important role in shaping voter opinions, providing valuable insights for campaign staff and managers. This project focuses on analyzing public sentiment by examining YouTube comments related to governor candidates in Puerto Rico during the three months leading up to the 2024 election. By analyzing and diving into these comments, the researcher aimed to uncover trends and patterns in voter sentiment, offering a unique perspective on how people viewed the candidates and their campaigns in each timeframe.

Using the TextBlob Python library, a reliable tool for processing textual data, the researcher conducted sentiment analysis to categorize comments into positive, negative, or neutral sentiments. The project combined modern client-side and server-side technologies to ensure a user-friendly system for managing and visualizing sentimental data. This approach streamlined the analysis process, making it both efficient and accurate for handling large datasets.

The investigation outlines the entire process, from data collection and preparation to the system's architecture and sentiment analysis methodology. By detailing the technologies and steps involved, including data cleaning and categorization, the project provides a comprehensive look at how public sentiment on digital platforms can influence political campaigns and their strategies going forward.

RELATED WORK

Social Media Sentiment Analysis Using Python

According to Bhumika Gupta's paper, sentiment analysis is a crucial tool in natural language processing (NLP) that extracts subjective opinions from text, particularly in social media platforms like Twitter [1]. It is widely used to classify sentiments into categories such as positive, negative, and neutral. There are two main approaches to sentiment analysis: lexicon-based and machine learning-based methods. Machine learning models, including Support Vector Machines (SVM) and neural networks, have proven effective in capturing complex sentiment patterns [1]. Preprocessing techniques, such as tokenization and stop word removal, play a critical role in improving classification accuracy. In the case of this project, the researcher uses TextBlob, a Python library that does sentimental analysis using the machine learning method.

One of the most significant applications of sentiment analysis is in politics. Political sentiment analysis is used to gauge public opinion on government policies, political parties, and election campaigns. In the case of Gupta's paper, by analyzing tweets, researchers and policymakers can track sentiment trends, identify voter concerns, and predict election outcomes [1]. The ability to process real-time data from Twitter allows for rapid insights into political discourse and helps politicians adjust their strategies based on public sentiment. As sentiment analysis techniques continue to evolve, their role in political monitoring and decision-making will likely expand, offering valuable insights into public opinion dynamics [1].

TextBlob Library

TextBlob is a Python library that significantly simplifies the process of sentiment analysis by offering a suite of tools designed to process textual data with ease. Among its many features, it supports tokenization, which involves breaking down text into smaller components such as words or sentences, and noun phrase extraction, which identifies and isolates meaningful phrases from text.

Additionally, TextBlob provides built-in functionality for polarity and subjectivity scoring. Polarity scoring measures the sentiment of the text on a scale from -1 (negative sentiment) to 1 (positive sentiment), with 0 being neutral sentiment. Subjectivity scoring assesses how subjective or opinion-based the text is, as opposed to being factual.

One of the standout advantages and the main reason why this library was chosen for this project is its intuitive and user-friendly API [2]. This design ensures that developers and data scientists can efficiently process text and integrate TextBlob's features with other Python libraries, making it a versatile tool for a wide range of tasks regarding natural language text. Whether used for simple sentiment analysis or as a component in a more complex application, TextBlob provides an accessible entry point for working with text data.

METHODOLOGY

Data Collection

The data was collected through Google video searches. First the investigator searched for the candidate and added "site:youtube.com" at the end to only include YouTube videos, as shown in Figure 1. Then filtered the results by month (August, September, October 2024), as shown in Figure 2. Finally, extracted video IDs located in the URL of each video. The IDs were organized into CSV files categorized by candidate and month, as shown in Figure 3. This structured approach ensures accurate and efficient data analysis.

Figure 1

YoutTube Exclusive Site Search

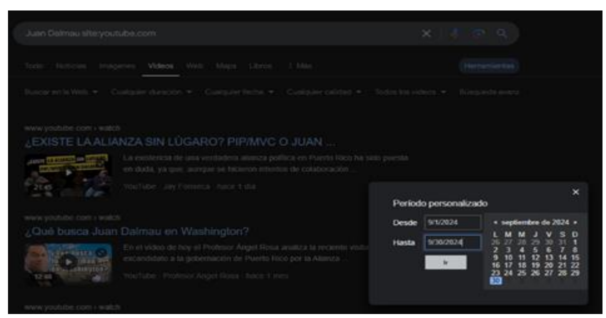

Figure 2

Date Range Search

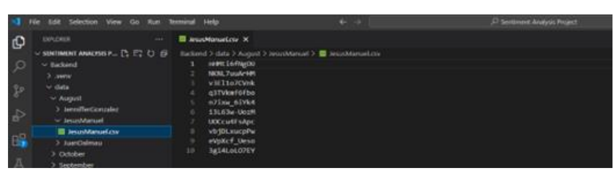

Figure 3

Video URL Ids CSV Files

System Architecture

Frontend: The client is a single-page application built using ReactJS [3]. It provides a user-friendly interface where users can select a candidate and a month from dropdown menus and initiate the analysis via a button, as shown in Figure 4. After clicking the Analyze button, it will deactivate and show a loading text, as shown in Figure 5. Finally, when the analysis ends, it will show the results below the input fields, as shown in Figure 6.

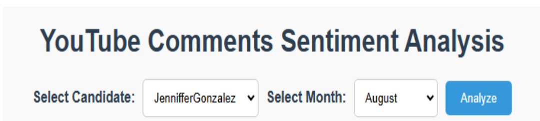

Figure 4

Front-End Client

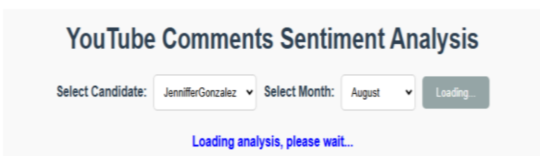

Figure 5

Front-End Client Loading

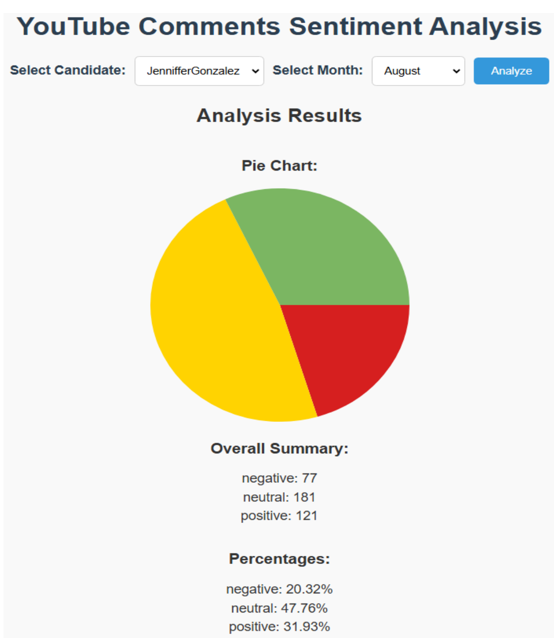

Figure 6

Front-End Client Result

Backend: The backend server is a REST API designed in the Python programming language [4] and implements a POST endpoint named 'analyze' that was implemented using the Flask framework [5]. When triggered, this endpoint processes the selected dataset, cleans the comments, and performs sentiment analysis using various libraries. The first thing made was to locate the YouTube video ids desired located in the data collected CSV files. The code for this is shown in Figure 7. After that the code loops through each of the ids in the selected file and connects to the google API [6] to extract the comments for each video and do the Sentiment Analysis on every one of them, as shown in Figure 8. To analyze the comments, the code must first translate the text to English. To do that the researcher wrote a function that detected the language of each comment and translated it, if necessary, before doing the sentiment analysis. The code to check the language and translating if necessary is shown on Figure 10. After that, the investigator used the TextBlob library to do the sentiment analysis of the comment [2]. Finally, the researcher used the result of the sentiment analysis to structure the data for the front-end client to present it as shown in Figure 9.

```
@app.route('/analyze/<candidate>/<month>', methods=['POST'])
@cross_origin(supports_credentials=True)
def analyze(candidate, month):
    youtube = googleapiclient.discovery.build(
        api_service_name, api_version, developerKey=DEVELOPER_KEY
    )

    # Path to the CSV file
    csv_file_path = f'data/{month}/{candidate}/{candidate}.csv'

    # Initialize overall sentiment summary
    overall_summary = {
        'positive': 0,
        'negative': 0,
        'neutral': 0
    }

    # Initialize word list for word cloud
    word_list = []

    # Read video IDs from the CSV file
    with open(csv_file_path, newline='') as csvfile:
        reader = csv.reader(csvfile)
        video_ids = [row[0] for row in reader]  # Assuming video I
```

Figure 7

Reading CSV File

```python
# Loop through each video ID
for video_id in video_ids:
    print(video_id)
    request = youtube.commentThreads().list(
        part="snippet",
        videoId=video_id,
        maxResults=100
    )

    try:
        response = request.execute()

        # Analyze sentiments for each comment
        for item in response.get('items', []):
            comment_text = item['snippet']['topLevelComment']['snippet']['textDisplay']
            sentiment = analyze_sentiment(comment_text)
            overall_summary[sentiment] += 1

            # Collect words for word cloud (convert to lowercase and split)
            word_list.extend(comment_text.lower().replace('.','').replace(',','').split())
    except:
        continue
```

Figure 8

Analyzing the sentiment of each YouTube comment

```python
# Calculate percentages
total_comments = sum(overall_summary.values())
percentages = {
    sentiment: (count / total_comments) * 100 if total_comments > 0 else 0
    for sentiment, count in overall_summary.items()
}

# Prepare data for pie chart
pie_chart_data = [
    {'label': sentiment, 'value': count}
    for sentiment, count in overall_summary.items()
]

# Prepare data for word cloud (word frequencies)
word_frequencies = dict(Counter(word_list))

# Combine all data into the response
response_data = {
    'overall_summary': overall_summary,
    'percentages': percentages,
    'pie_chart_data': pie_chart_data,
    'word_frequencies': remove_from_keys(word_frequencies)
}

return jsonify(response_data)
```

Figure 9

Setting up result data

```python
def analyze_sentiment(text):
    try:
        language = detect(text)
    except:
        language = 'en'

    if language != 'en':
        if len(text) < 5000:
            text = GoogleTranslator(source='auto', target='en').translate(text)

    analysis = TextBlob(text)

    if analysis.sentiment.polarity > 0:
        return 'positive'
    elif analysis.sentiment.polarity == 0:
        return 'neutral'
    else:
        return 'negative'
```

Figure 10

Language translation code

ANALYSIS RESULTS

The following tables are the results of the sentiment analysis, made with the methodology previously mentioned, of the three candidates for Governor of Puerto Rico in the three months leading up to the 2024 Puerto Rican election. These results were compared with each other to analyze the sentiment trajectory of each candidate leading up to the election in the results discussion section.

Table 1

August Sentiment Analysis

Candidate	Positive	Negative	Neutral
Jenniffer Gonzalez	31.66%	20.32%	48.02%
Juan Dalmau	36.98%	22.32%	40.70%
Jesus Manuel Ortiz	42.23%	15.14%	42.63%

Table 2

September Sentiment Analysis

Candidate	Positive	Negative	Neutral
Jenniffer Gonzalez	31.21%	18.26%	50.53%
Juan Dalmau	37.36%	17.55%	45.09%
Jesus Manuel Ortiz	45.60%	13.47%	40.93%

Table 3

October Sentiment Analysis

Candidate	Positive	Negative	Neutral
Jenniffer Gonzalez	35.28%	17.89%	46.82%
Juan Dalmau	40.44%	11.63%	47.92%
Jesus Manuel Ortiz	42.57%	18.07%	39.36%

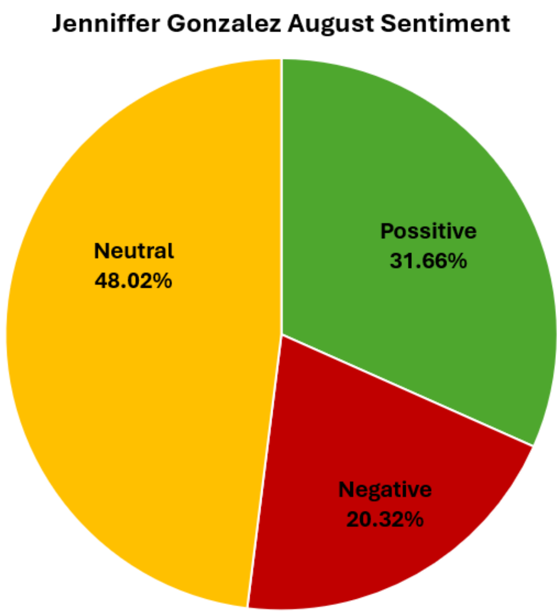

Figure 11

Jenniffer Gonzalez August Sentiment

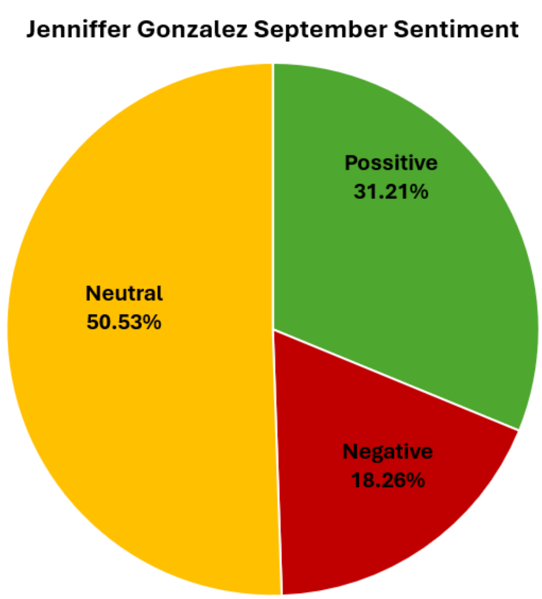

Figure 12

Jenniffer Gonzalez September Sentiment

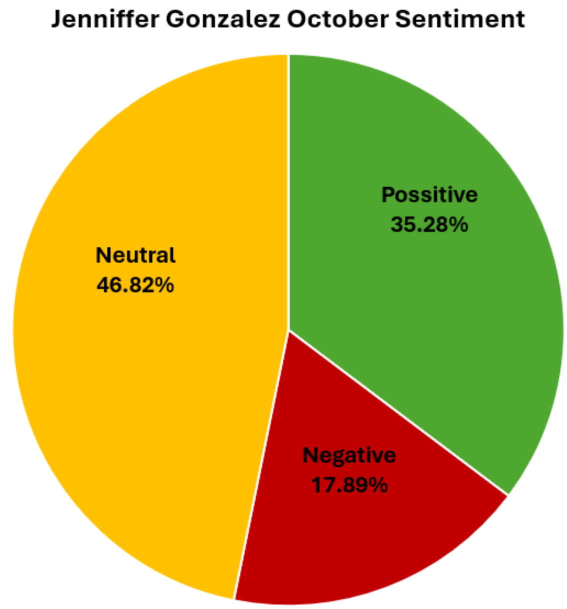

Figure 13

Jenniffer Gonzalez October Sentiment

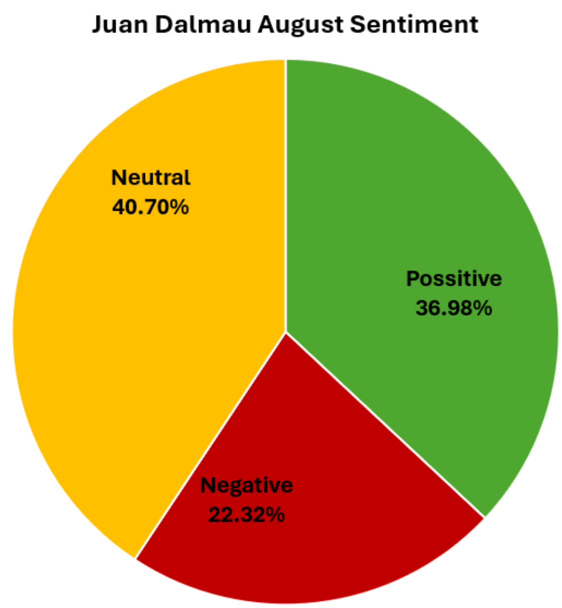

Figure 14

Juan Dalmau August Sentiment

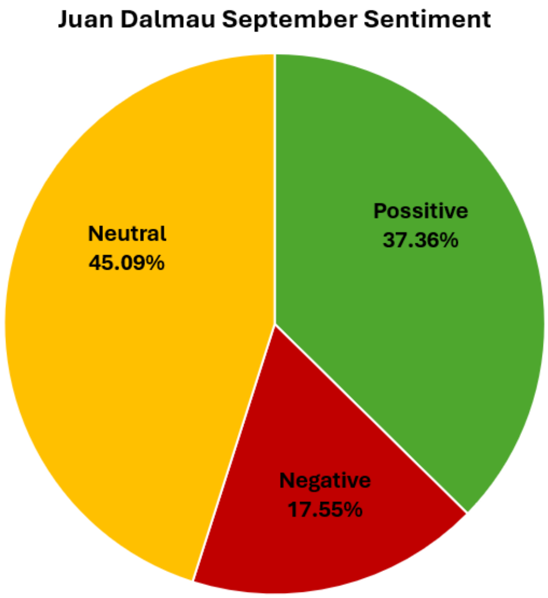

Figure 15

Juan Dalmau September Sentiment

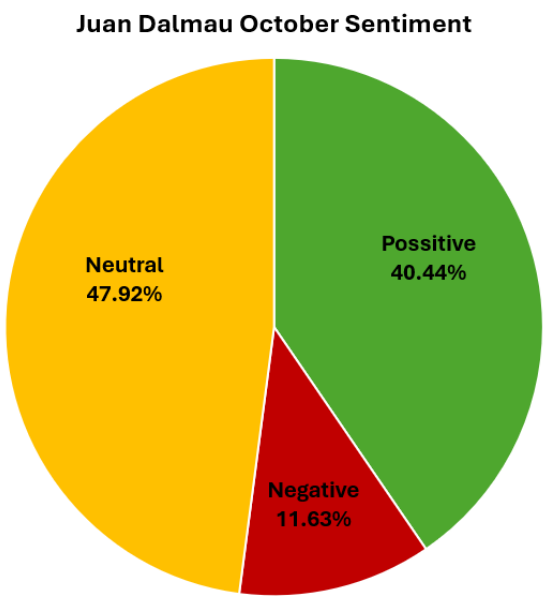

Figure 16

Juan Dalmau October Sentiment

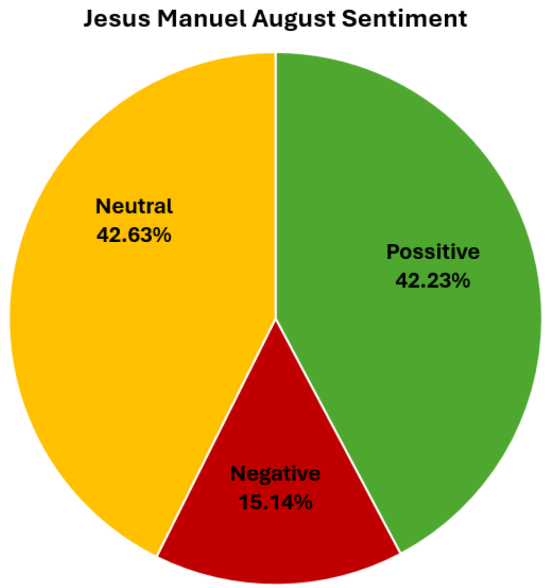

Figure 17

Jesus Manuel August Sentiment

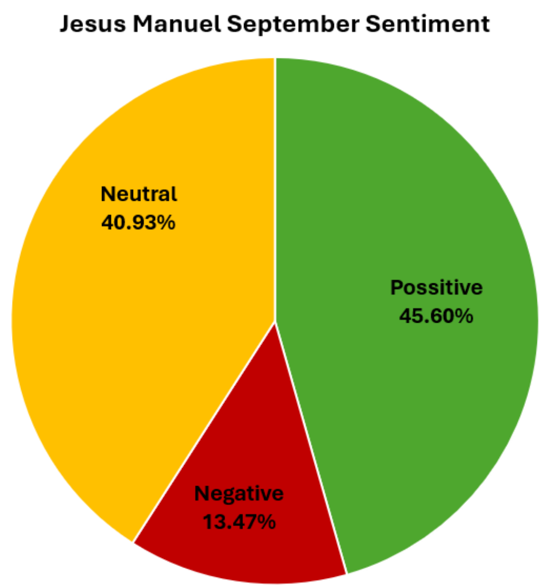

Figure 18

Jesus Manuel September Sentiment

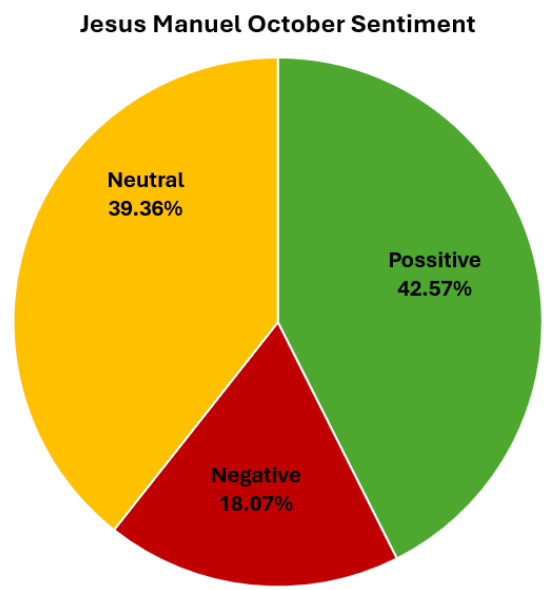

Figure 19

Jesus Manuel October Sentiment

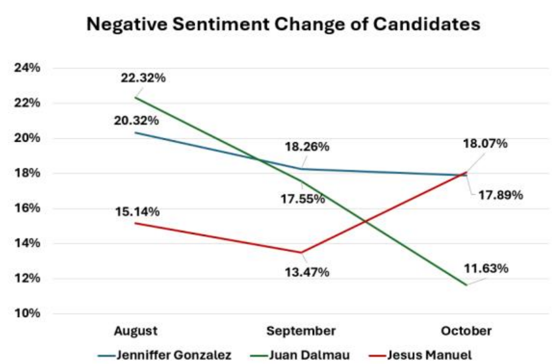

Figure 20

Negative Sentiment Change of Candidates

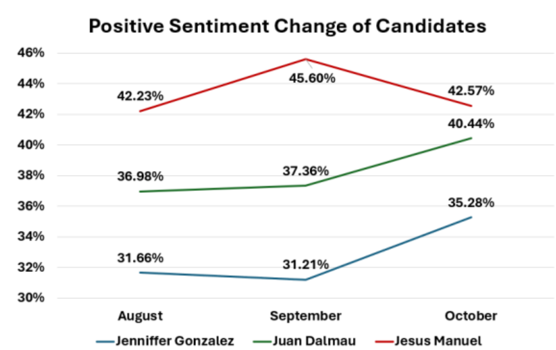

Figure 20

Positive Sentiment Change of Candidates

Discussion

Jenniffer González started in August with a relatively balanced sentiment distribution, with 31.66% positive, 20.32% negative, and 48.02% neutral. In September, her positive sentiment dropped slightly to 31.21%, and her negative sentiment fell to 18.26%, while neutral sentiment rose to 50.53%, suggesting the public was becoming less polarized and more indifferent about her. By October, González regained positive sentiment, rising to 35.28%, while negative sentiment decreased further to 17.89%. Neutral sentiment fell slightly to 46.82%, indicating a growing favorable perception and a decline in criticism, as more people shifted away from neutrality.

Juan Dalmau showed consistent improvement across all months. In August, his sentiment was 36.98% positive, 22.32% negative, and 40.70% neutral. By September, his positive sentiment increased to 37.36%, while negative sentiment fell significantly to 17.55%, and neutral sentiment rose to 45.09%. This indicates a decline in criticism and growing appeal. In October, Dalmau's positive sentiment jumped further to 40.44%, with negative sentiment dropping to 11.63% and neutral sentiment rising to 47.92%. This steady increase in favorability and reduction in criticism highlights his growing momentum as a leading candidate.

Jesús Manuel Ortiz consistently maintained high levels of positive sentiment, starting in August with 42.23% positive, 15.14% negative, and 42.63% neutral. In September, his positive sentiment increased to 45.60%, with negative sentiment slightly declining to 14.47%, while neutral sentiment dropped to 40.93%. This indicates polarization, with more people shifting from neutral to positive perceptions. However, in October, Ortiz's positive sentiment fell slightly to 42.57%, and negative sentiment rose to 18.07%, while neutral sentiment remained stable around 39.36%. This suggests some emerging criticism or controversies.

Overall, Juan Dalmau emerged as the candidate with the most consistent improvement, particularly in reducing negative sentiment and increasing positive views, making him the most positively perceived candidate in October, taking into consideration the data collected from YouTube comments. Jenniffer González showed signs of recovery after a very neutral phase, with a notable improvement in positive sentiment toward the end of the period. Jesús Manuel Ortiz, while maintaining strong overall favorability, faced a slight rise in negative sentiment in October, suggesting some potential issues at the end of his campaign leading to the election.

Conclusion

This project shows how modern web development tools and powerful sentiment analysis libraries can work together to dig into the political sentiment around specific candidates. Using the TextBlob library along with other frameworks and technologies, the researcher was able to handle large datasets of comments, uncovering interesting patterns and trends in how people feel about candidates. Specifically, this project analyzed the sentiment trajectory for each candidate based on YouTube comments, which gave the researcher a good sense of how public opinion shifted over time on this platform.

There's room to make this tool even better. In the future, others, including the researchers, could expand the analysis to include more social media platforms like X (formerly known as Twitter) and Facebook. These platforms have a wider variety of conversations and interactions that could give different people a more accurate picture of voter sentiment. Of course, doing this would require more time, resources, and a bigger budget to access and process all that extra data.

With these upgrades, this tool could become even more precise and provide deeper insights into how people feel about political candidates at a given timeframe across multiple platforms.

References

[1] B. Gupta, M. Negi, K. Vishwakarma, G. Rawat, and P. Badhani, "Study of Twitter Sentiment Analysis using Machine Learning Algorithms on Python," *Int. J. Comput. Appl.*, vol. 165, pp. 29–34, 2017. [Online]. Available: https://doi.org/10.5120/ijca2017914022. [Accessed: February 1, 2025].

[2] Loria, S. (2024, 02, 15). TextBlob: Simplified Text Processing. TextBlob: [Online]. Available: https://textblob.readthedocs.io/en/dev/. [Accessed: January 6, 2025].

[3] Meta Platforms, Inc. (2024, 12, 05). React v19. [Online]. Available: https://react.dev/blog/2024/12/05/react-19. [Accessed: January 6, 2025].

[4] Python Software Foundation. (2025, 12, 03). Python 3.13.1 documentation. Python: [Online]. Available: https://docs.python.org/3/. [Accessed: January 6, 2025].

[5] Pallets. (2024, 11, 13). Flask Framework Documentation: *Pallets*. [Online]. Available: https://flask.palletsprojects.com/en/stable/. [Accessed: January 6, 2025].

[6] Google for Developers. (2022, 07, 21). YouTube Data API v3: Google for Developers: [Online]. Available: https://developers.google.com/youtube/v3/docs. [Accessed: January 6, 2025].