Election Pulse: Unveiling Voter Sentiments for 2024 US Election Forecasting

Team members:Hemanth Kumar Reddy Gunnam, Supraja Pericherla, Durga Pravallika Kuchipudi

Introduction

This project endeavors to harness the vast online data and advanced sentiment analysis techniques to predict the outcome of the 2024 elections, providing a cutting-edge approach to understanding and quantifying public opinion in real-time. In an era where traditional polling falls short, our sentiment analysis aims to capture the nuanced, dynamic public sentiment that proliferates across social media and online forums, offering a more immediate and granular insight into voter trends and preferences. By tapping into the digital discourse that increasingly influences political landscapes, this project not only reflects the current digital zeitgeist but also marks a pivotal shift towards data-driven methodologies in forecasting election dynamics.

Dataset Description

Our project utilizes a dataset strategically mined from Reddit and Google to forecast the 2024 election outcomes through sentiment analysis. From Reddit, we extracted a wealth of user-generated content, including posts and comments from politically active subreddits, which offers a grassroots-level view of voter sentiment and discourse. Google's contribution to our dataset comprises an array of news headlines, search trends, and public reactions, providing a broad perspective on the political climate. This meticulously compiled dataset combines the nuanced, in-depth discussions from Reddit with the expansive, diverse sentiments reflected in Google's data, setting the stage for a comprehensive sentiment analysis that aims to capture the multifaceted political pulse of the nation.

Flowchart:

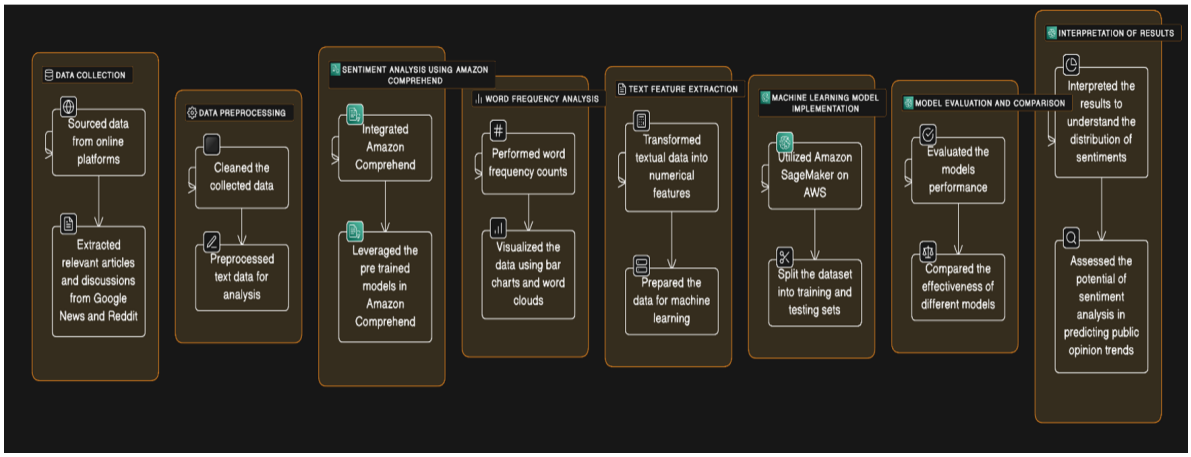


Fig 1: Flow chart of the methods

The flowchart in the image outlines our sentiment analysis project methodology in a structured sequence of steps. Starting with data collection, we sourced relevant articles and discussions from online platforms like Google News and Reddit, then moved to data preprocessing where the collected data was cleaned and preprocessed for analysis. We integrated Amazon Comprehend for sentiment analysis, leveraging its pretrained models, and performed word frequency analysis, visualizing the data with bar charts and word clouds. The textual data was transformed into numerical features for machine learning, utilizing Amazon SageMaker, and the dataset was split into training and testing sets. We concluded with model evaluation and comparison to assess performance, ultimately interpreting the results to understand sentiment distribution and assess the potential of sentiment analysis in predicting public opinion trends.

Dataset

Data Mining:

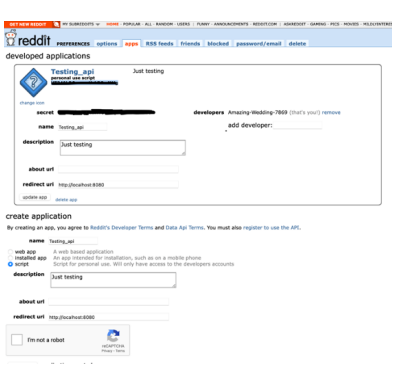
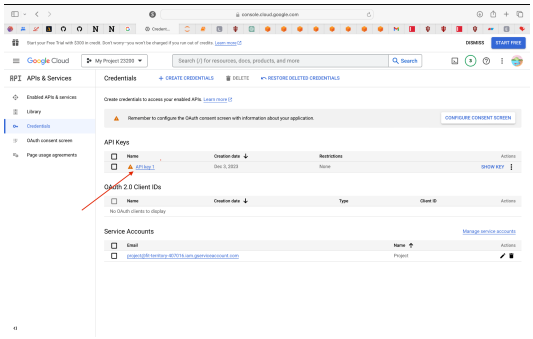


Fig 2 :Reddit API



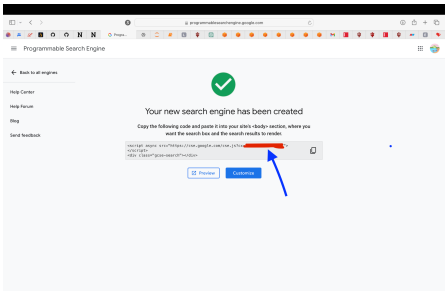


Fig 3: Google news API

In our project targeting the 2024 US elections, we developed a web scraping script using the PRAW library to mine the subreddit 'politics' for recent submissions specifically related to 'US elections 2024', employing the 'lucene' syntax for sophisticated search capabilities. Concurrently, we set up a Google Cloud Console project to create and configure a secure API key and a customized Programmable Search Engine, enabling us to crawl relevant content. Leveraging Python, requests, and pandas, we scripted a robust data extraction process from Google News, querying multiple election-related topics. The script fetched and parsed data across ten pages, meticulously aggregating details like titles, links, and snippets into a structured dataset. This data was then transformed into a pandas DataFrame, and subsequently exported to a CSV file for comprehensive analysis, creating an efficient pipeline for the collection and preparation of a substantial dataset aimed at understanding public sentiment towards the upcoming elections.

Data preprocessing:

1. **Data Cleaning**: Unnecessary columns, such as 'Body', are removed from the dataset, and relevant columns, 'Title' and 'Comments', are combined to form a single text field for analysis.
2. **Tokenization and Stopword Removal**:

❖ Utilized the NLTK library to tokenize the text and eliminate common English stop words. However, during the analysis of the top words, it was observed that some of the common words persisted. Consequently, an alternative approach was employed by defining common stop words and removing them from the dataset.

Calculating Sentiment Polarity for the Dataset using AMAZON S3 & AMAZON COMPREHEND:

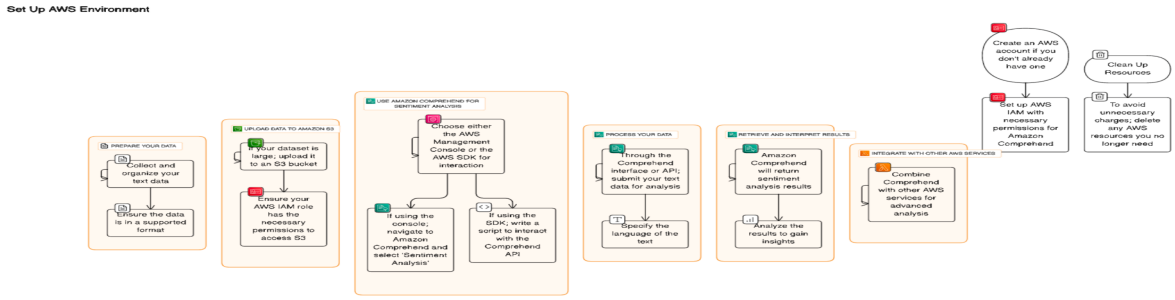
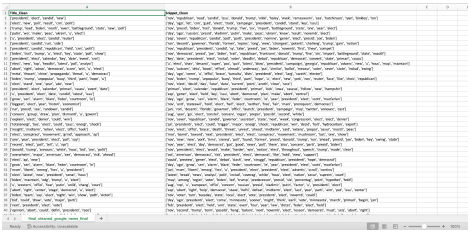


Fig 4: Cloud architecture

In our project, we initiated the data preparation phase by gathering and organizing text data for compatibility and secure storage in an Amazon S3 bucket. We then set up AWS IAM roles to ensure secure data access, a vital step for maintaining data integrity within AWS. For sentiment analysis, we leveraged AWS Comprehend via the Management Console or the AWS SDK, utilizing its machine learning capabilities for accurate sentiment detection and analysis. Post-analysis, we reviewed the results for actionable insights and integrated them with other AWS services for enhanced depth and insight. The methodology also included efficient AWS resource management, involving setting up necessary AWS accounts, configuring IAM for Comprehend, and prudent resource cleanup to maintain cost-effectiveness.

Sentiment Analysis Results Using Amazon Comprehend



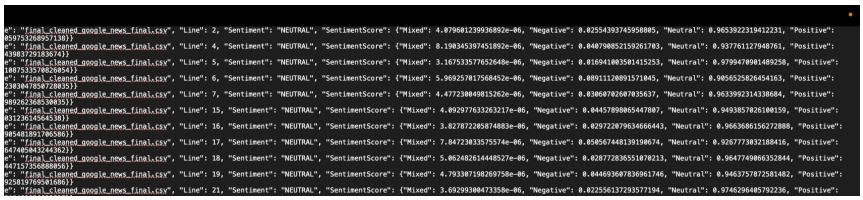


Fig 5:Sentiment scores

Using Amazon Comprehend NLP, our project processed a precompiled dataset, employing sophisticated algorithms to classify text into 'Positive,' 'Negative,' 'Neutral,' and 'Mixed' sentiments. The analysis primarily identified a neutral sentiment, occasionally punctuated by positive or negative tones, providing nuanced emotional insights into each text string. These sentiment scores enrich the dataset, offering valuable applications in market research and social media analysis. This refined tool allows organizations to assess public opinion and consumer sentiment with enhanced depth.

Results:

Sentiment score distribution:

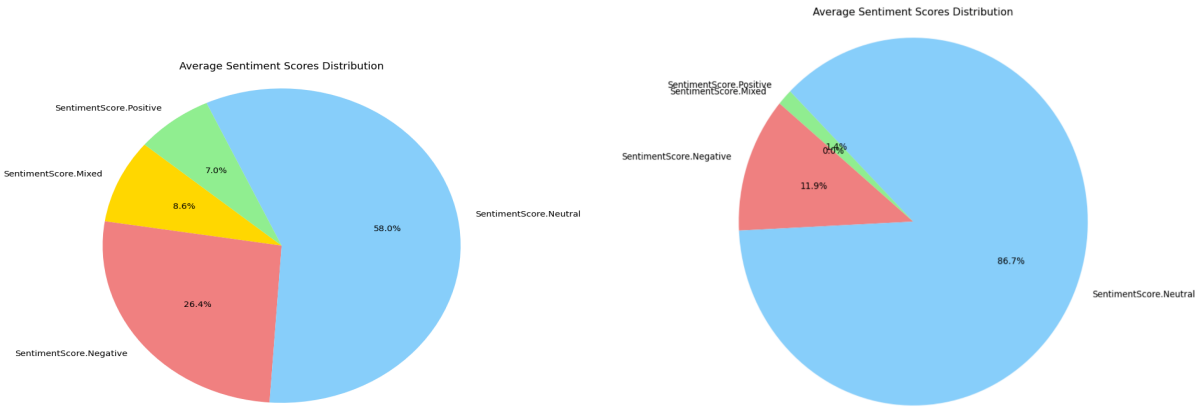


Fig 6: Pie chart of reddit and google news

Using Amazon Comprehend for sentiment analysis, our project processed data, assigning sentiment scores (positive, negative, neutral, or mixed) through machine learning. The tabular representation and a pie chart visualization showed a predominant neutrality at 86.7%, suggesting a dataset rich in factual language. Negative sentiment was at 11.9%, positive at 0.4%, and mixed emotions at 1.0%. This underscores the dataset's largely neutral tone, offering insights for content strategies and market analysis within the Google dataset.

Top 10 words analysis:

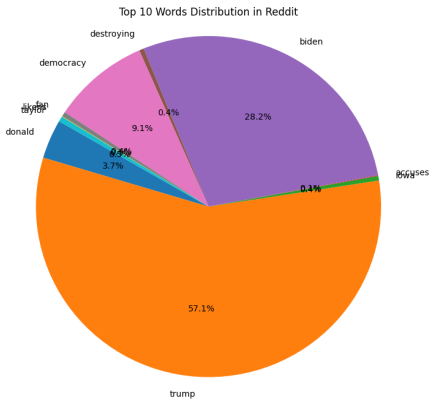


Fig 7: Pie chart displaying top 10 words in Reddit

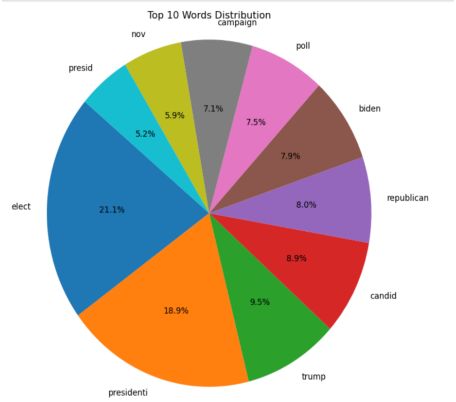


Fig 8: Pie chart of top 10 words in google

The analysis of the top 10 words in Reddit and Google News datasets revealed distinct word frequency distributions, illustrated through pie charts. In the Reddit dataset, the term "trump" significantly dominates, constituting 57.1% of the chart, emphasizing its prevalence. Conversely, the Google News dataset's chart highlights 'elect' and 'presidenti' at 21.1% and 18.9%, suggesting a focus on election-related themes, with additional terms like 'republican', 'campaign', 'poll', and 'nov'. This variation signifies a political emphasis within the Google dataset, contrasting with Reddit's focus on the individual figure of Trump. These visual representations offer insights into the unique political discourse prevalent on each platform.

Word Rank:

	Word	Frequency	Rank
0	donald	178	237.0
1	trump	2748	1.0
2	iowa	21	1869.0
3	accuses	3	6586.0
4	biden	1357	6.0

Fig 9:Word Rank of Reddit

	Word	Frequency	Rank
0	elect	360	1.0
1	presidenti	322	2.0
2	trump	161	3.0
3	candid	151	4.0
4	republican	136	5.0

Fig 10:Word Rank of Google

The Reddit dataset analysis highlights a strong focus on political figures, particularly Donald Trump and Joe Biden, with "trump" being the most dominant word, indicating a politically oriented discourse. In contrast, the Google dataset shows a varied political lexicon, with 'elect' and 'presidenti' leading in frequency, pointing to a broader focus on electoral processes and political entities. Words like 'candid' and 'republican' in the Google dataset suggest a rich spectrum of election-related content, differing from Reddit's more concentrated emphasis on individual political figures.

Word Cloud:

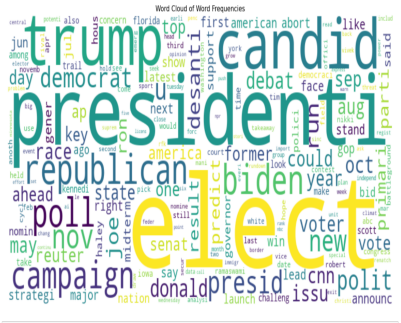


Fig 11:Word cloud of Google

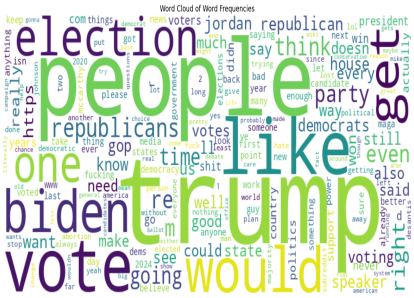


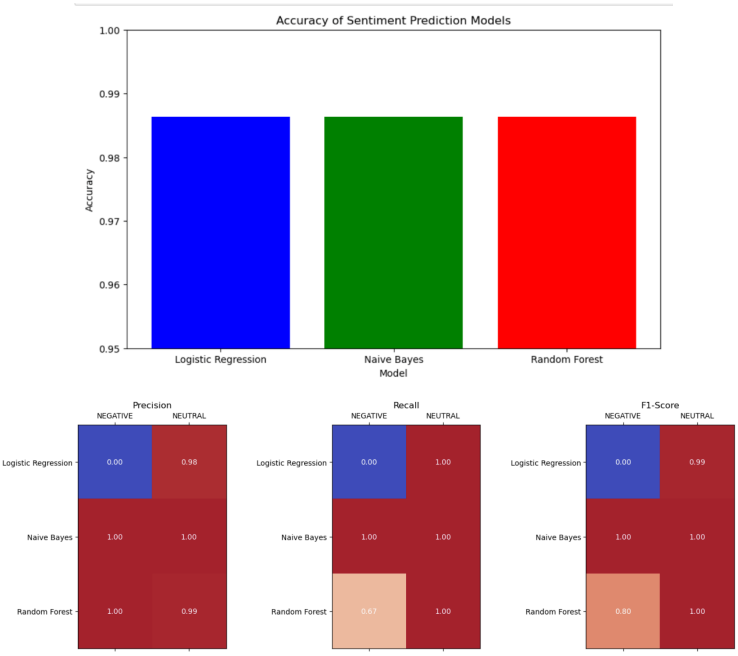
Fig 12:Word cloud of Reddit

Word clouds from Reddit and Google datasets reveal prevalent themes in 2024 U.S. election discussions, with "Trump" being a highly frequent term in both. The Google dataset's cloud emphasizes a diverse political vocabulary, showcasing terms like "President," "Democrat," "Republican," and "Biden," along with "vote" and "election," highlighting a focus on electoral activities and political figures. These visualizations succinctly encapsulate key topics and sentiments, offering insights into the dominant political discourse across both platforms.

Machine learning models:

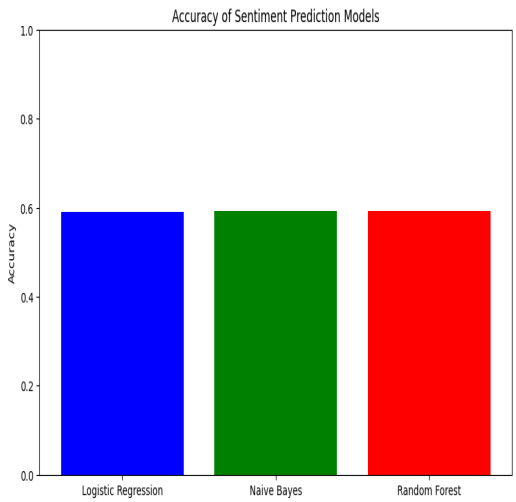
For Google:

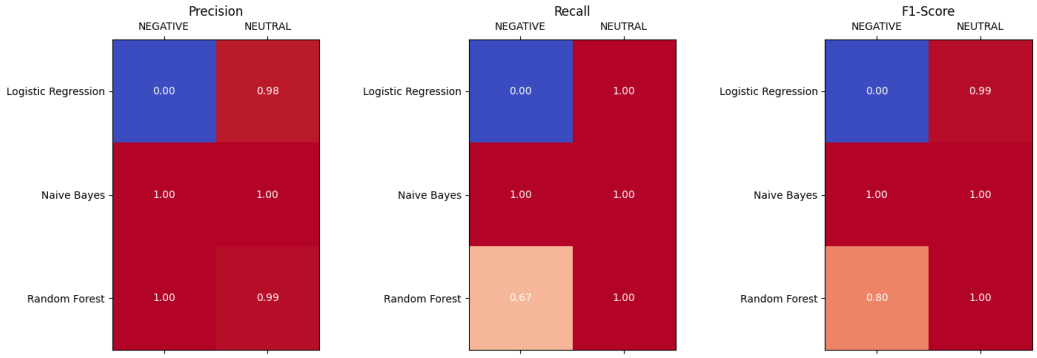
```
Cross-Validated Accuracy - Logistic Regression: 0.9834094368340944
Cross-Validated Accuracy - Naive Bayes: 0.9834094368340944
Cross-Validated Accuracy - Random Forest: 0.9834094368340944
Classification Report - Logistic Regression:
              precision    recall  f1-score   support

    NEGATIVE       0.00      0.00      0.00         6
     NEUTRAL       0.98      1.00      0.99       355

    accuracy                           0.98       361
   macro avg       0.49      0.50      0.50       361
weighted avg       0.97      0.98      0.98       361

Classification Report - Naive Bayes:
              precision    recall  f1-score   support

    NEGATIVE       1.00      1.00      1.00         6
     NEUTRAL       1.00      1.00      1.00       355

    accuracy                           1.00       361
   macro avg       1.00      1.00      1.00       361
weighted avg       1.00      1.00      1.00       361

Classification Report - Random Forest:
              precision    recall  f1-score   support

    NEGATIVE       1.00      0.67      0.80         6
     NEUTRAL       0.99      1.00      1.00       355

    accuracy                           0.99       361
   macro avg       1.00      0.83      0.90       361
weighted avg       0.99      0.99      0.99       361
```



For Reddit:

```
Cross-Validated Accuracy - Logistic Regression: 0.6120182576333739
Cross-Validated Accuracy - Naive Bayes: 0.5945450810324535
Cross-Validated Accuracy - Random Forest: 0.6105921134716603
Classification Report - Logistic Regression:
              precision    recall  f1-score   support

       MIXED       1.00      0.14      0.25       770
    NEGATIVE       1.00      0.09      0.16      3152
     NEUTRAL       0.62      1.00      0.77      6720
    POSITIVE       1.00      0.09      0.17       575

    accuracy                           0.64     11217
   macro avg       0.90      0.33      0.34     11217
weighted avg       0.77      0.64      0.53     11217

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatio
  _warn_prf(average, modifier, msg_start, len(result))
Classification Report - Naive Bayes:
              precision    recall  f1-score   support

       MIXED       1.00      0.02      0.04       770
    NEGATIVE       0.72      0.04      0.08      3152
     NEUTRAL       0.61      1.00      0.76      6720
    POSITIVE       0.00      0.00      0.00       575

    accuracy                           0.61     11217
   macro avg       0.58      0.27      0.22     11217
weighted avg       0.64      0.61      0.48     11217

Classification Report - Random Forest:
              precision    recall  f1-score   support

       MIXED       0.99      0.14      0.25       770
    NEGATIVE       0.99      0.09      0.16      3152
     NEUTRAL       0.62      1.00      0.77      6720
    POSITIVE       1.00      0.09      0.17       575
```





While analyzing Google and Reddit datasets, Logistic Regression, Naive Bayes, and Random Forest models achieved high accuracies of 98.3%. However, they exhibited varied performances. Logistic Regression struggled with 'NEGATIVE' sentiments, Naive Bayes raised overfitting concerns with perfect scores, and Random Forest showed high precision but lower recall for 'NEGATIVE.' Naive Bayes performed best despite potential overfitting, Logistic Regression faced challenges with class imbalance, and Random Forest achieved a balance between precision and recall. The bar chart visually contrasts outcomes, emphasizing the importance of model selection in sentiment analysis tasks.

Conclusion:

In conclusion, our sentiment analysis project successfully applied Logistic Regression, Naive Bayes, and Random Forest models to a Google dataset, revealing each model's strengths and weaknesses in the context of political sentiment classification. The Naive Bayes model demonstrated exceptional performance, though it raised concerns about overfitting, which will be a focus for improvement. Logistic Regression struggled with class imbalance, pointing to the need for more sophisticated data preprocessing techniques. Random Forest offered a balanced

performance, but there is room to enhance its recall for negative sentiments. This study underscores the complexity of sentiment analysis within the dynamic domain of political discourse and sets the stage for future advancements in machine learning methodologies. The insights gained provide valuable feedback for refining algorithms and improving the accuracy of sentiment prediction—a crucial tool for understanding and forecasting political trends.

Discussion:

In our project's discussion, we observed that while Naive Bayes achieved high accuracy, its potential overfitting to the Google dataset highlights the intricacies of model selection and the necessity for diverse training data. Logistic Regression's limited success in classifying negative sentiments underscores the challenges linear models face with nuanced textual data, suggesting a need for more sophisticated approaches or balanced datasets. The Random Forest model, with its balanced performance yet room for improvement in recall, points towards the potential benefits of further feature engineering. These findings emphasize the critical nature of algorithmic adaptability to the subtleties of human language, especially in politically sensitive contexts, and the need for ongoing refinement of models to accurately interpret the fluid landscape of online discourse.

Reference:

https://github.com/chaithanya21/Sentiment-Analysis-using-Pyspark-on-Multi-Social-Media-Data/blob/master/SENTIMENTAL_ANALYSIS_OF_CUSTOMER_PRODUCT_REVIEW.ipynb

Appendix: Contribution from Each member

Durga Pravallika

- Google Dataset Mining and flowchart diagram
- Word cloud, word rank , word analysis and rank distribution
- ML analysis and results
- Final report editing

Supraja Pericherla-

- Worked to extract data from Reddit API
- Worked on Amazon comprehend and S3 to get sentiment scores
- Performed ML analysis for reddit dataset

Hemanth

- Dataset research
- Methodology
- Discussions
- Conclusion