Presidential Election Forecasting based on Sentiment Analysis using Twitter Data

Lohith Prasanna Teja Kakumanu
Industrial and Systems Engineering
Data Analytics
University at Buffalo, State University
of New York
Buffalo, NewYork, USA
lohithpr@buffalo.edu

***Abstract*— This study outlines a method for monitoring public opinion on presidential competition in the 2020 U.S. election in real-time as it was expressed on the blogging platform Twitter. People now primarily share their thoughts about political parties and candidates via Twitter. Approximately one-quarter of American adults use Twitter, and when they convey their opinions on the platform, they mostly discuss political issues and politicians. A new Pew Research Center analysis of English-language tweets made by a representative sample of U.S. adult Twitter users between May 1, 2020, and May 31, 2021 reveals that one-third (33%) of those tweets are political in character. An upsurge in Twitter activity frequently follows breaking news or emerging events, offering a rare chance to measure the relationship between public opinion and election outcomes. "Without the tweets, I wouldn't be here… I have over 100 million followers between Facebook, Twitter (and) Instagram," Trump told the Financial Times in 2017. "Over 100 million. I don't have to go to the fake media." This clearly shows that public opinion can implicate user's social and political perspectives. We will apply data science methodologies to examine the impact of twitter's public comments on the 2020 U.S Presidential Elections.**

In this research we use Sentiment analysis to investigate how these events impact public opinion. We use Machine Learning Algorithms like Random Forest, Decision Tree, Logistic Regression, Support Vector Machines, Naive Bayes, and NLP Techniques to generalize the Sentiment analysis Model to gain the maximum insights in terms of understanding the computational linguistics to identify, analyze, extract the sentiments or emotions in the responses. As part of this extraction and cleaning the data there is a possibility of loss of information due to the vectorization of the words so we use Valence Aware Dictionary and sEntiment Reasoner (VADER), Transformer Architecture BERT with Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) for a comprehensive analysis of text data. This model effectively captures syntactic, semantic, and contextual information while also performing sentiment analysis.

Keywords— Presidential Election, Twitter data, Sentiment Analysis, Text Mining, NLP, Deep Learning, BERT, ROBERTA, CNN, VADER, Textblob, Spacy.

I. INTRODUCTION

It's a matter of fact that US presidential election is one of the pivotal political event which grabs the cynosure of the world as it is one of the crucial event for keeping up the democracy. This celebration of democracy started when George Washington became the president in the first presidential election during 1787-1789. However, there is minutia about the presidential election such as Electoral College where the President is not directly elected by the popular vote. Instead, the Electoral College system is used. Each state has a certain number of electors based on its representation in Congress (the total number of its senators and representatives). The candidate who wins the popular vote in a state generally receives all of that state's electoral votes, except in Maine and Nebraska, which use a proportional allocation method. The electors chosen by each state's voters cast their votes for President. This process occurs in mid-December, with the President of the Senate presiding over a joint session of Congress to count the electoral votes.

A. Social Media

Social media, particularly Twitter, played an important role in the rapid delivery of news, information, and campaign updates. During the election season, candidates, campaigns, and news outlets used the platform to share real-time updates, making it a primary source for breaking news. It promoted conversations about critical election issues and allowed people to express thoughts and perspectives. They may communicate their policy positions, react to events in real time, and interact with voters and supporters. This direct communication provided unfiltered access to the messages of the candidates. Organizations and campaigns also used Twitter to educate voters about registration procedures, voting techniques, and important election dates. This contributed to citizens having access to critical information about the political process. To combat the spread of disinformation during the election, Twitter deployed fact-checking techniques. Given the proliferation of inaccurate or misleading material on social media sites, this was especially critical. Furthermore, Twitter enabled users to react in real time to big events such as presidential elections, and trends and hashtags relating to these events provided a snapshot of public emotion and reactions

As a result, this sparked the concept of forecasting public feelings or emotions from the Twitter platform on the 2020 US Presidential Elections using Natural Language Processing (NLP), a subfield of Data Science and AI.

B. Sentiment Analysis

So, Sentiment Analysis is a specific Application of NLP that focuses on interpreting the sentiment or emotional tone indicated in a piece of text, such as a review, comment, or social media post. It entails identifying the text as positive, negative, or neutral, and sometimes further categorizing it into finer-grained sentiment categories; it can be classified at the sentence-level [1], the document-level, and the word or aspect-level. The complete document is viewed as a single entity for document-level categorization, whereas a sentence may be regarded as a mini-document for sentence-level categorization. Aspect-based SA focuses on a single aspect or word and its related polarity. Similarly, the task of SA comprises essentially four steps main: preprocessing, feature extraction, classification, and interpretation of results, among

various domains like movie reviews, election opinion prediction, airline reviews, amazon reviews, etc. Among all the steps mentioned above, feature extraction plays an important role for improving the classification efficiency. There are two types of feature extraction methods, such as lexicon-based methods and machine learning-based or deep learning-based methods [2].

C. Approach

In this research, the Twitter public opinion on 2020 US presidential elections dataset from the Kaggle have been collected and I have started the feature extraction and feature engineering involving steps like tweets preprocessing and cleaning. Some processes that could be conducted for tweets preprocessing are: removing twitter handles (@user); removing punctuation, numbers and special characters; removing short words lemmatization, tokenization, stemming, vectorization. And, then the cleaned data in the required format can be passed to the Machine Learning models, neural network architectures which is one of the largest model in the NLP which has a large number of parameters, large size, and high latency. Now, the Accuracy of all the models were observed based on the training and validation data set and the better performed model is picked and unseen data will be passed for the prediction of the Presidential Election.

II. LITERATURE REVIEW

Similar studies are available using different approaches in the same area. Sentiment analysis, preliminary analysis, and data collection tools are often applied to these studies. A summary of past studies is briefly described in this section and implications from this information are summarized provided one of the initial studies on social media data and election result prediction for German parliamentary elections. The study reported that Twitter was widely used for political conversion and that these social media conversations were reflective of offline political landscapes [3]

In [4], the authors developed a new method for semantic knowledge extraction from research documents and article using an integration of semantic technology, NLP, and information extraction.

In this paper Miftahul et.al [5] collected the data from Kaggle and the collected data was pre-processed by removing URLs, converting emojis to text, removing stop words, stemming and lemmatization, removing collection words, and tokenization. The authors proposed 5 classification algorithms such as linear support vector classification, RF, Decision Tree(DT), Logistic Regression, and Naive Bayes to predict and classify the textual data from Twitter about the 2020 US election campaign. Future research may focus on evaluating methods differences to determine accuracy levels. Simulating various proportions of fake tweets into a dataset might give insight into counterfeit tweets' effect on model performance. This study, however, shows that the 2020 election could be predicted in favor of Joe Biden by analyzing Twitter sentiment a month before the election.

Similarly, Ethan Xia et.al [6] in their study they analyzed the twitter data for the SA of the 2020 U.S. Presidential Election they used methodologies like Support Vector Machine, Decision Tree Classifier, Gaussian Naïve Bayes, AdaBoost Classifier, and Multi-Layer Perceptron Classifier. "The best classifier, MLP, was used to analyze our collected tweets that are related to the 2020 U.S. election. This demonstrates that social media data can be utilized as an alternative tool to predict election results. Beyond this, they also discovered some interesting phenomena, such as the important role of negative sentiment within social media and the correlation between political events and sentiment trends".

Another study, A large-scale sentiment analysis of tweets pertaining to the 2020 US presidential election by Rao Hamza Ali et.al [7] used NLTK, TextBlob, CoreNLP, and spaCy for preprocessing. They manually analyzed subsets of tweets to verify the sentiment score assigned to them by Valence Aware Dictionary and sEntiment Reasoner (VADER). Except for a few cases where sarcasm was not detected, they found VADER making accurate sentiment classifcations. Through the use of sentiment analysis and groupings of different users and their tweets, they provided useful insight into how political conversations, at a time of major political events, are conducted wholly on a social platform like Twitter.

[8] A study by Quratulain Alvi1 et.al provided a detailed analysis of existing sentiment classification techniques in chronological order and categorizes them into statistical, lexicon, oncology, supervised, unsupervised, and deep learning approaches. They concluded that deep learning approach produced promising results. Also, they have analyzed that while there may be observed correlations between specific Twitter trends or sentiment patterns and election outcomes, "it does not necessarily imply that these correlations indicate a causal relationship or direct influence on the election results."

A study by Hao Wang et.al [9] for Real-time Twitter Sentiment Analysis of 2012 U.S. Presidential Election Cycle they have used the Twitter "firehose" and expert-curated rules and keywords to get a full and accurate picture of the online political landscape using sentiment App from Amazon Mechanical Turk their architecture and methodology was generic the future scope can be extended to other domains like system for gauging sentiments about films and actors surrounding Oscar nomination and selection.

Hassan Nazeer et.al [10] performed Sentiment Analysis on the 2020 before and after US Elections twitter data set their research employed TF-IDF to extract features from the given tweet and used a Naive Bayes classifier to obtain positive or negative sentiment for the given candidate. In most cases, the public opinion expressed over Twitter coincided with the election results, except four outliers. To summarize, for all states where sentiment results did not corroborate with election results, long-term trends before and after the election reveal that there was an increase in the positive sentiment of the winning candidate. They could have used other Algorithms or some deep learning techniques for getting better insights.

[11] Md. Rakibul Hassan et.al, have developed a model to sentiment analysis which allows the processing of Twitter API streaming feed in real time and to classify its polarity to provide valuable insight. They have used Bag of Words (BoW) and TF- IDF for the vectorization and finally built some classifier which can be utilized as data analysis tools in NLTK.

Widodo Budiharto et.al [12], in their study they have focused on tweets data related to 2019 Presidential election with top keywords. They created their own approach to calculate the polarity and the score They calculated the score based on the difference between the number of positive and negative words and polarity by dividing score by total number of sentiment words.

[13] Xiaokang Gong et.al, in their study they performed Sentiment Analysis To alleviate the reliance on annotated data that combines data augmentation techniques and Transformer is proposed.

Sayyida Tabinda Kokab et.al [14] in their research, they used enhanced feature extraction and classification model using BERT model and dilated convolutional Bi-LSTM model. A BERT-based CBRNN SA model has been proposed for sentence-level classification. The proposed hybrid CBRNN model was applied on four diverse domain datasets namely US-airline reviews, self-driving car reviews, US-presidential election reviews, and movie reviews. The performance of the CBRNN model is evaluated using five statistical measures, such as accuracy, precision, f1-score, recall, and AUC. As a future direction, the proposed technique can be applied to other resource-poor languages. Furthermore, another future direction is to implement the model on multi-class classification problems.

III. METHODOLOGY

A. *Data Collection*

I have collected the data from the Kaggle source [API: kaggle datasets download -d manchunhui/us-election-2020-tweets]. In this collection manchunhui scooped the data from twitter. He mentioned in the Kaggle notebook that He tapped into Twitter's vault using the Twitter API and a cool feature called statuses_lookup. Focused on gathering tweets related to the U.S. presidential election. The original intent was to update the dataset daily. To widen the net, he used snscrape to fish for tweets based on specific words. This approach broadened the scope of data collection, ensuring a more comprehensive dataset. As part of the time frame considerations, he initially aimed to cover the timeframe between October 15, 2020, and November 4, 2020. Later he extended the collection until at least the end of November 6, 2020, with ongoing election events. He planned further updates with the goal of covering tweets until the end of November 8, 2020.[US Election 2020 Tweets (kaggle.com)]

B. *Exploratory Data Analysis*

Explored the structure and contents of the datasets. Conducted an initial examination of missing values and replaced the numerical values with mean and categorical with mode. As a result of the initial examination its found that the Trump dataset shows an average of 7.48 likes and 1.70 retweets per tweet, with users having an average of 22,603 followers. Geographical coordinates indicate diverse locations. The Biden dataset has a slightly higher average of 10.16 likes and 2.13 retweets, with users averaging 28,850 followers. Geographical patterns are consistent with variability within specific ranges.

Conducted Exploratory Data Analysis to answer the questions such as how likes were distributed across the two datasets, how the distribution of likes and user follower counts was presented, how the number of tweets was distributed to discern prevailing trends, what the top 10 tweet sources were, and how many tweet counts were present concerning the top 5 countries associated with Trump and Biden hashtags.

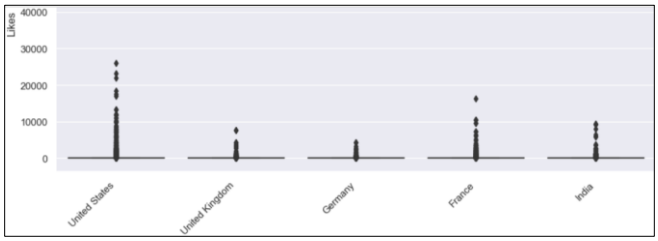


Fig: 1 Distribution of Likes on the Trump Dataset on top 5 countries

The above boxplot illustrates the distribution of Twitter likes across different countries, with the United States leading in likes, followed by the United Kingdom, Germany, France, and India. Each box represents the middle 50% of data, with the median marked inside. While the U.S. exhibits substantial variation (IQR of 10,000), India's variation is smaller (IQR of 2,500), indicating diverse likes within each country. Outliers are observed in France (35,000 likes) and India (1,000 likes), suggesting exceptional data points beyond the typical range.

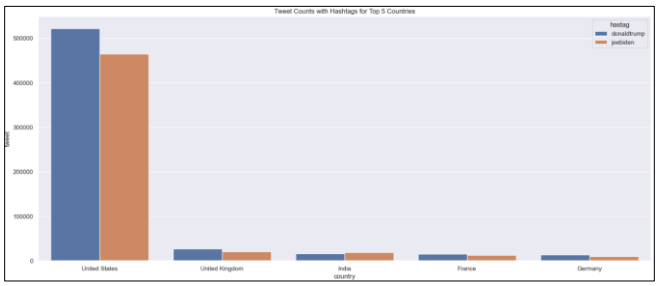


Fig: 2 Distribution of Tweet counts with Hashtags for top5 countries

The tweet count graph illustrates that the United States dominates Twitter discussions for both Donald Trump and Joe Biden, with the U.S. having the highest tweet count followed by the United Kingdom, India, France, and Germany. Notably, Donald Trump outpaces Joe Biden in tweet popularity across all countries, with the U.S. showing a significant margin—Trump exceeding 500,000 tweets and Biden just over 400,000. This aligns with Trump's more

active and polarizing Twitter presence, tweeting frequently throughout the day, compared to Biden's less frequent engagement. The graph hints at Trump's Twitter popularity across the listed countries.

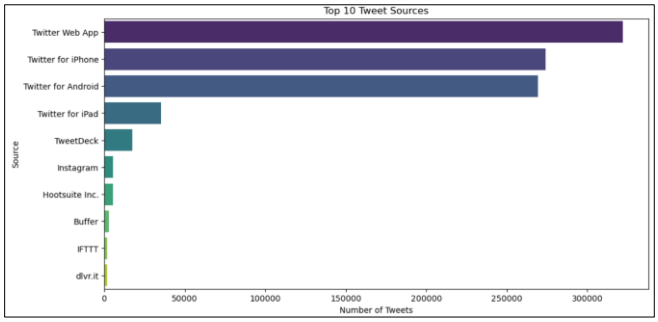


Fig 3: Top 10 tweet sources

The above visual representation depicts the top 10 tweet sources based on tweet volume, with Twitter Web App leading the list, followed by Twitter for iPhone, Twitter for Android, and Twitter for iPad. Notably, Twitter dominates as the primary tweet source, indicating widespread use of the Twitter app on mobile devices. Additionally, Tweet Deck, a popular Twitter management tool, is prominently utilized for tweeting. Social media management tools such as Instagram, Hootsuite Inc., Buffer, IFTTT, and dlvr.it also contribute to tweet publishing. The overall trend suggests a growing reliance on social media management tools for efficient tweet scheduling and publication, reflecting a desire to streamline the tweeting process.

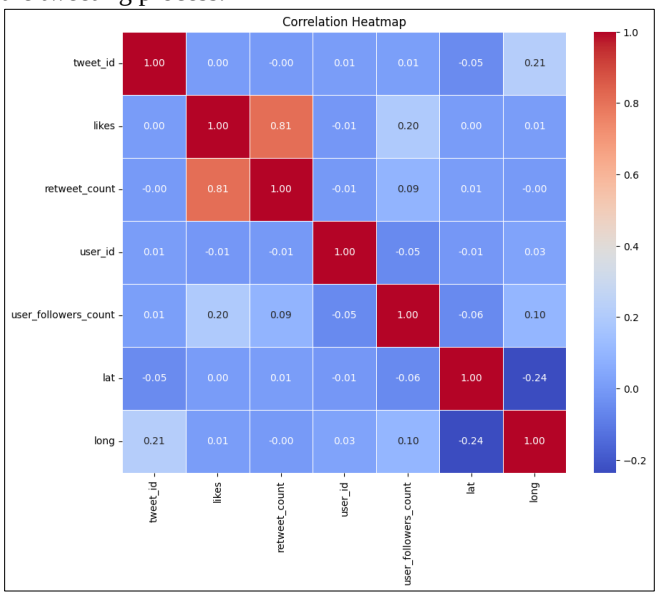


Fig 4: Correlation HeatMap of Numerical Variables

The correlation heatmap explores associations among key variables, revealing strong positive correlations between likes and retweet_count (0.81) and user_followers_count and likes (0.20). This suggests tweets with more likes tend to receive additional retweets, and users with larger followings attract heightened engagement. Conversely, weak negative correlations (e.g., user_id and likes, lat and retweet_count) hint at subtle tendencies, such as similar engagement levels for tweets from the same user. Caution is advised in interpreting these weak correlations. Overall, the heatmap underscores a positive link between likes, retweets, and user followers, indicating more engaging tweets are likely to be visible, particularly for users with larger followings.

C. Text Preprocessing

The text preprocessing for tweets related to Donald Trump and Joe Biden was demonstrated, employing the Natural Language Toolkit (nltk) library. The preprocess_text function, utilizing nltk, transformed text to lowercase, tokenized it, removed stopwords, and applied stemming with the Porter Stemmer. Processed tweets were stored in 'processed_tweet' columns, contributing to enhanced data cleanliness for past analyses such as sentiment analysis or topic modeling.

i. Converting all text to lowercase ensures uniformity in the data. This is crucial because text analysis and machine learning models often treat "Word" and "word" as different entities. Converting everything to lowercase helps avoid such distinctions, ensuring consistent representation.

ii. Performed tokenization to break down a text into individual units, providing a structural understanding of the composition, be it words, phrases, or symbols.

iii. After tokenization, common English stopwords are removed from the text using the set of stopwords from NLTK. This step helps eliminate common words that don't contribute much to the analysis

iv. After removing stop words I have performed stemming, where each token (word) is reduced to its root or base form using the Porter Stemmer. This process helps in reducing words to their base form for better analysis and interpretation.

These all operations prepares the text data for natural language processing (NLP) tasks. Clean and tokenized text is often a prerequisite for building machine learning models or conducting meaningful analyses. But before proceeding with any model building vectorization is required where each of the processed text is converted to numeric which will be fed to the model for the training with the respective response variable.

D. Sentiment Analysis:

Sentiment refers to the emotional tone expressed in text, categorized as positive, negative, or neutral. Sentiment analysis, a natural language processing technique, assesses and extracts these emotional nuances from textual data. It's widely used to gauge public opinion, customer feedback, and attitudes toward specific topics or products. In my research I have tried to get the sentiment of the collected tweet to understand public opinion on of the top social media platform like twitter which contains the discussions on US presidential election with hashtags of Trump and Biden.

a. TexBlob Sentiment Analyzer: it is a free and open-source Python library that simplifies common Natural Language Processing (NLP) tasks. It provides a simple and intuitive API for tasks such as:

Sentiment analysis: Detecting the positive, negative, or neutral sentiment of a piece of text.
Part-of-speech tagging: Identifying the grammatical role of words (e.g., noun, verb, adjective).
Noun phrase extraction: Identifying the key nouns and noun phrases in a sentence.
Spelling correction: Correcting spelling errors in text.
Language detection: Identifying the language of a piece of text.
Text translation: Translating text between different languages.

The overall sentiment of the text is computed by averaging the polarity scores of all the words.

Mathematically, it can be represented as:

Sentiment = ($\sum$ Polarity_Scores) / Number_of_Words

TextBlob is built on top of other popular NLP libraries like NLTK and Pattern, making it easy to use even for those with limited NLP experience. It is also highly customizable and can be extended to perform more complex NLP tasks.

Here are some of the key use cases for TextBlob:

- Social media analysis: Analyzing the sentiment and opinions expressed in social media conversations about brands, products, or events.
- Customer reviews analysis: Understanding customer feedback and identifying common themes and complaints in customer reviews.
- Market research: Analyzing online discussions and reviews to understand consumer preferences and market trends.
- Spam detection: Identifying spam messages by analyzing their language patterns and content.
- Data cleaning and processing: Preprocessing and cleaning text data for other NLP tasks.
- Machine learning: TextBlob can be used as a feature engineering tool for machine learning models that involve text data.

b. VADER:

VADER (Valence Aware Dictionary and sEntiment Reasoner) is a Python library specifically designed for sentiment analysis of social media text. It offers several advantages:

- Lexicon and rule-based: VADER utilizes a lexicon containing words with associated sentiment scores, along with rules to handle sarcasm, negation, and intensifiers, resulting in a more nuanced understanding of sentiment than purely lexicon-based approaches.
- Specifically attuned to social media: With its focus on social media, VADER accounts for informal language, slang, and emojis, leading to more accurate sentiment analysis in online contexts.
- Open-source and easily accessible: VADER is freely available under the MIT license and readily accessible through PyPI, making it easy to integrate into Python projects.
- Provides sentiment intensity: VADER goes beyond simple positive/negative classification by offering a score representing the intensity of the detected sentiment, giving more insight into the overall sentiment of a text unit.

The normalization used by Hutto is:

$$\frac{x}{\sqrt{x^2+\alpha}}$$

where x is the sum of the sentiment scores of the constituent words of the sentence and alpha is a normalization parameter that we set to 15.

The normalization is graphed below:

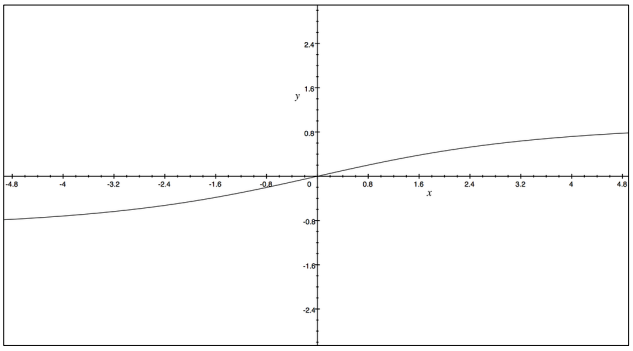


Fig: 5 Normalization graph

We see here that as x grows larger, it gets more and more close to -1 or 1. To similar effect, if there are a lot of words in the document you're applying VADER sentiment analysis to, you get a score close to -1 or 1. Thus, VADER sentiment analysis works best on short documents, like tweets and sentences, not on large documents.

Outcomes of the sentiments from the TextBlob and Vader were fed into the datasets. For our further analysis of modelling, we have to pass the sentiment as our dependent variable and the preprocessed text as independent variable and predict the unseen data from the biden and trump data sets.

E. Handling Imabalanced Dataset

We are combining the two data sets here for modelling because the modelling should not be biased towards any candidate. Also, Checked the distribution of sentiment labels and noticed that our sentiment labels were playing favorites. So, performed undersampling to balance 1s and 0s, and some reshuffling to make sure each sentiment label got its fair share of attention.

F. Vectorization

Once, the data set is balanced perfectly then we split the data into test and train and perform the vectorization. Vectorization techniques, such as TF-IDF or word embeddings, involve creating representations of words or

phrases based on the entire dataset. If you vectorize the entire dataset before splitting, information from the test set can leak into the training set, leading to optimistic performance estimates.
After splitting the data into 80-10-10 ratio then vectorization is applied. TF-IDF Vectorization is followed in the current research methodology.

It measures how important a term is within a document relative to a collection of documents (i.e., relative to a corpus). Words within a text document are transformed into importance numbers by a text vectorization process. There are many different text vectorization scoring schemes, with TF-IDF being one of the most common. As its name implies, TF-IDF vectorizes/scores a word by multiplying the word's Term Frequency (TF) with the Inverse Document Frequency (IDF).
Term Frequency: TF of a term or word is the number of times the term appears in a document compared to the total number of words in the document.

$$TF = \frac{\text{number of times the term appears in the document}}{\text{total number of terms in the document}}$$

Inverse Document Frequency: IDF of a term reflects the proportion of documents in the corpus that contain the term. Words unique to a small percentage of documents (e.g., technical jargon terms) receive higher importance values than words common across all documents (e.g., a, the, and).

$$IDF = log(\frac{\text{number of the documents in the corpus}}{\text{number of documents in the corpus contain the term} + 1})$$

Translated into plain English, importance of a term is high when it occurs a lot in a given document and rarely in others. In short, commonality within a document measured by TF is balanced by rarity between documents measured by IDF. The resulting TF-IDF score reflects the importance of a term for a document in the corpus.
This score is fed to the model for the training.

G. Modelling

NAÏVE BAYES: I have used the Naïve Bayes Approach for modelling. Naive Bayes classifiers are simple and computationally efficient. Naive Bayes works well for text classification tasks, including sentiment analysis. It has been widely used for spam filtering and sentiment analysis applications due to its effectiveness in dealing with high-dimensional and sparse feature spaces, which are common in text data.

One of the main reason to choose Naïve Bayes approach is due to its probabilistic nature which allows us for understanding how each feature (word) contributes to the overall sentiment prediction. And Naive Bayes can serve as a good baseline model. It provides a benchmark for more complex models to surpass.

Other Modelling Techniques included Random Forest and hyper parameter tuning using greidsearchcv and using Transformer.

TRANSFORMERS: Tried leveraging the Hugging Face Transformers library to integrate a sentiment analysis model into the analysis of processed tweets stored in a Pandas DataFrame. First, the sentiment analysis model, based on BERT architecture, is loaded using the AutoModelForSequenceClassification and AutoTokenizer modules from the Transformers library. The sentiment analysis pipeline is then created using the loaded model and tokenizer through the pipeline function from the Transformers library. Subsequently, the sentiment analysis is applied to each processed tweet in the 'processed_tweet' column of the DataFrame. The results, including sentiment labels and scores, are extracted into new columns ('sentiment_label' and 'sentiment_score'). Finally, the updated DataFrame, now enriched with sentiment analysis outcomes, is displayed. This approach showcases the powerful capabilities of the Hugging Face Transformers library for seamlessly incorporating advanced natural language processing models into sentiment analysis tasks on textual data.

NEURAL NETWORK: Constructed a simple neural network for binary classification using a bag-of-words approach with CountVectorizer.
Implementation:

- The sentiment classes are label-encoded using LabelEncoder.
- The data is split into training and testing sets using train_test_split
- The 'processed_tweet' column is vectorized using CountVectorizer. This step converts the text data into numerical features, representing the word counts for each document (tweet).

Neural Network Architecture: The neural network consists of two layers:

- Input Layer: Dense layer with 64 neurons and ReLU activation function.
- Output Layer: Dense layer with 1 neuron and Sigmoid activation function (for binary classification).

RELU Activation Function: It introduces non-linearity to the model, allowing it to learn complex patterns. The function itself is quite simple and is defined as follows:

ReLU(x) = max (0, x)

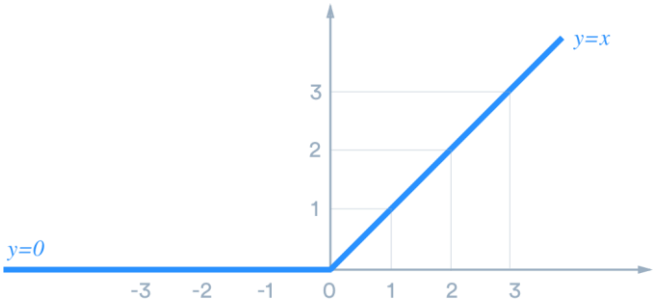


Fig: 6 RELU Curve

Since ReLU gives output zero for all negative inputs, it's likely for any given unit to not activate at all which causes the network to be sparse. Now let us see how ReLu activation function is better than previously famous activation functions such as sigmoid and tanh.

The activations functions that were used mostly before ReLU such as sigmoid or tanh activation function saturated. This means that large values snap to 1.0 and small values snap to -1 or 0 for tanh and sigmoid respectively. Further, the functions are only really sensitive to changes around their mid-point of their input, such as 0.5 for sigmoid and 0.0 for tanh. This caused them to have a problem called vanishing gradient problem. Let us briefly see what vanishing gradient problem is.

Neural Networks are trained using the process gradient descent. The gradient descent consists of the backward propagation step which is basically chain rule to get the change in weights in order to reduce the loss after every epoch. It is important to note that the derivatives play an important role in updating of weights. Now when we use activation functions such as sigmoid or tanh, whose derivatives have only decent values from a range of -2 to 2 and are flat elsewhere, the gradient keeps decreasing with the increasing number of layers.

This reduces the value of the gradient for the initial layers and those layers are not able to learn properly. In other words, their gradients tend to vanish because of the depth of the network and the activation shifting the value to zero. This is called the vanishing gradient problem.

ReLU, on the other hand, does not face this problem as its slope doesn't plateau, or "saturate," when the input gets large. Due to this reason models using ReLU activation function converge faster [16].

The model is compiled using the Adam optimizer and binary cross-entropy loss function. Accuracy is chosen as the evaluation metric.

Adam Optimizer: Adam is an adaptive learning rate optimization algorithm that's been designed specifically for training deep neural networks. First published in 2014, Adam was presented at a very prestigious conference for deep learning practitioners - ICLR 15. It combines ideas from two other popular optimization algorithms, namely RMSprop and Momentum. Adam maintains both per-parameter learning rates and momentum-like terms for each parameter.

Algorithm:

Step 1: Initialization: Initialize the first and second moments to zero. These are vectors m and v with the same length as the parameter vector θ.

Step 2: Set the time step $t = 0$

Step 3: Parameter Updates: For each iteration t:

Compute the gradient g_t of the objective function with respect to the parameters θ_t. Update the first moment estimate m_t and the second moment estimate v_t as follows:

$$mt = \beta_1 \cdot m_{t-1} + (1-\beta_1) \cdot g_t$$
$$vt = \beta_2 \cdot v_{t-1} + (1-\beta_2) \cdot g_t 2$$

Bias correction for moment estimates:

$$m\hat{}t = mt / (1-\beta_1 t)$$

$$v\hat{}t = v_t / (1-\beta_2 t)$$

Update the parameters:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \varepsilon}} \cdot \hat{m}_t$$

β1 and β2 are the hyperparameters controlling the exponential decay rates.

η is the learning rate.

ε is a small constant to avoid divide by zero error [15]

The output Layer is Dense layer with 1 neuron and Sigmoid activation function

$$y = \sigma(W_2 \, . \, h_1 + b_2)$$

Where σ is the Sigmoid function, W_2 is the weight matrix, h_1 is the output from the hidden layer, and b_2 is the bias vector

This architecture forms a fully connected feedforward neural network for binary classification

H. Model Evaluation:

The trained model was evaluated on the unseen data and sentiment scores were calculated. The respective correlation matrix and Accuracy score for the test set were observed.

The Test Accuracy for the Naïve Bayes Approach resulted as 0.76. The Naive Bayes approach in my sentiment analysis model achieved an accuracy of 76.12%.

```
Test Accuracy of Naive Bayes Approach: 0.761227398758152

Confusion Matrix:
 [[61562 40866]
 [ 8048 94380]]
```

The confusion matrix reveals 61,562 true negatives, 40,866 false positives, 8,048 false negatives, and 94,380 true positives.

```
Classification Report:
              precision    recall  f1-score   support

           0       0.88      0.60      0.72    102428
           1       0.70      0.92      0.79    102428

    accuracy                           0.76    204856
   macro avg       0.79      0.76      0.75    204856
weighted avg       0.79      0.76      0.75    204856
```

Fig: 7 Model evaluation metrics

In terms of precision, recall, and f1-score, class 0 (negative sentiment) has values of 0.88, 0.60, and 0.72, while class 1 (positive sentiment) has values of 0.70, 0.92, and 0.79. The overall macro-averaged precision is 0.79, recall is 0.76, and f1-score is 0.75. The weighted average precision, recall, and f1-score are also 0.79, 0.76, and 0.75, respectively.

In Case of Neural Network the Accuracy was very impressive:

```
25607/25607 [==============================] - 31s 1ms/step - loss: 0.0491 - accuracy: 0.9916
Train Accuracy: 0.9916306734085083
Test Accuracy: 0.98785001039505
```

The neural network model achieved a high accuracy of 99.16% on the training dataset and an impressive accuracy of 98.79% on the test dataset. This indicates the model's strong performance in accurately classifying the data.

I. Analysis on sentiment Output

The output from the VADER and TEXTBLOB sentiment Scores were then mapped to the individual data sets and the distribution of sentiments were observed.

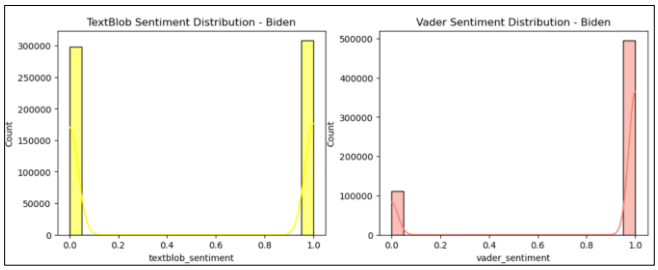


Fig: 8 Distribution of sentiment

The above graphs illustrate the sentiment score distributions generated by two distinct sentiment analysis algorithms, namely TextBlob and Vader. In the TextBlob sentiment distribution, the prevalent sentiment scores cluster between 0.2 and 0.4, with a central peak around 0.3. This distribution implies that the majority of the text scrutinized by TextBlob tends to exhibit a neutral or mildly positive sentiment. On the other hand, the Vader sentiment distribution mirrors a similar pattern, with the most frequent sentiment scores residing between 0.2 and 0.4. Notably, the Vader distribution manifests a slightly elevated peak, approximately at 0.35, indicating a higher propensity for Vader to identify text as positive compared to TextBlob.

The comparative analysis underscores the shared inclination towards neutral or mildly positive sentiments in the text scrutinized by both methods. The nuanced difference lies in Vader's heightened likelihood to categorize text as positive, a characteristic that may be attributed to its training on social media data, which often embodies a more emotionally expressive language compared to other text genres.

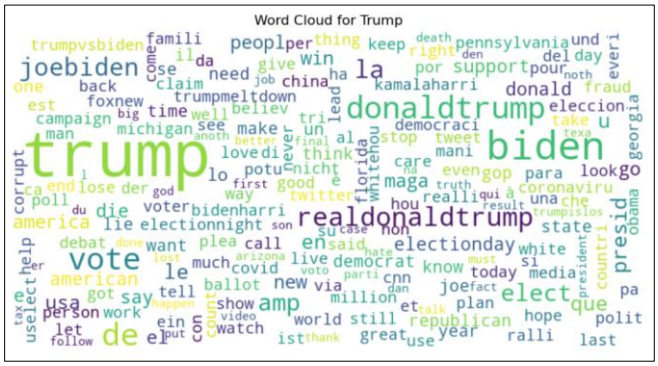


Fig: 9 The word cloud for the updated data set

The above word cloud is a visualization of the most common words in a collection of text about Donald Trump and Joe Biden from the hastagged Trump Dataset . The size of each word in the word cloud corresponds to its frequency in the text.

The largest word in the word cloud is "Trump", which suggests that he is the most frequently mentioned person in the text. Other prominent words in the word cloud include "Biden", "family", "supporters", "policies", and "tweet". This suggests that the text is primarily about Donald Trump and his presidency, with a focus on his family, supporters, policies, and use of Twitter.

Here are some other notable words in the word cloud:

- Positive
  words: "great", "win", "love", "care", "truth"
- Negative
  words: "fraud", "election", "meltdown", "lose", "co rrupt"
- Neutral
  words: "people", "time", "way", "know", "state"

The word cloud also includes the names of several states, such as "Pennsylvania", "Michigan", and "Georgia". These are all states that were key to the outcome of the 2020 presidential election, which suggests that the text may also discuss the election itself.

Overall, the word cloud provides a high-level overview of the main topics that are discussed in the text about Donald Trump and Joe Biden. It is clear that Trump is the most frequently mentioned person in the text, and that the text is primarily about his presidency and the 2020 election.

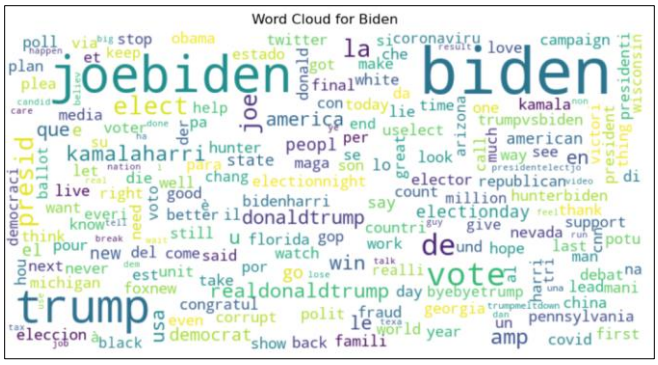


Fig: 10 The Wordcloud for the JoeBiden Data set

The above word cloud is from Joe Bidens Dataset. The most prominent words in the word cloud are "Joe Biden", "president", and "election". This suggests that the word cloud is focused on Biden's presidency and his election victory in 2020.

Other prominent words in the word cloud include:

- Policies: "plan", "care", "jobs", "climate", "econom y"
- People: "Kamala Harris", "voters", "Americans", "people", "families "
- Events: "election", "inauguration", "COVID-19", "debate", "speech"
- Media: "Twitter", "Fox News", "CNN", "MSNBC", "New York Times"

These words suggest that the word cloud is also focused on Biden's policies, the people he represents, and the events that have shaped his presidency. The prominence of media outlets in the word cloud suggests that the media plays a significant role in shaping public perceptions of Biden's presidency.

Overall, the word cloud suggests that Biden's presidency has been focused on the following:

- His policies, such as his plan for the economy, his care for Americans, and his commitment to jobs, climate, and the economy
- The people he represents, such as voters, Americans, people, and families
- The events that have shaped his presidency, such as the election, inauguration, COVID-19 pandemic, debates, and speeches
- The media coverage of his presidency

It is important to note that a word cloud is just one way to interpret Biden's presidency. There are many other ways to interpret his presidency, and the word cloud may not reflect all of the perspectives on his presidency. However, the word cloud does provide a valuable snapshot of some of the key themes and issues associated with Biden's presidency.

J. Approach to achieve goal of the Problem Statement

Attempted to analyze the sentiments extracted from tweets and mapped them to the overall responses to comprehend the aggregate public sentiment. I reviewed the output sentiment counts obtained from both VADER and TextBlob analyses.

```
print('trump',df_trump['vader_sentiment'].value_counts())
print('trump',df_trump['textblob_sentiment'].value_counts())
print('biden',df_biden['vader_sentiment'].value_counts())
print('biden',df_biden['textblob_sentiment'].value_counts())
```

```
trump vader_sentiment
1    520437
0    168840
Name: count, dtype: int64
trump textblob_sentiment
1    346306
0    342971
Name: count, dtype: int64
biden vader_sentiment
1    494943
0    111819
Name: count, dtype: int64
biden textblob_sentiment
1    308283
0    298479
Name: count, dtype: int64
```

The above data presents the sentiment counts for tweets related to Trump and Biden, as determined by two different sentiment analysis algorithms: Vader and TextBlob.

For Trump-related tweets analyzed by Vader, the count indicates 520,437 instances classified as positive sentiment (1) and 168,840 instances classified as negative sentiment (0). Meanwhile, TextBlob's analysis of Trump-related tweets yielded 346,306 instances with positive sentiment (1) and 342,971 instances with negative sentiment (0).

In the case of Biden-related tweets, Vader identified 494,943 instances with positive sentiment (1) and 111,819 instances with negative sentiment (0). On the other hand, TextBlob's analysis of Biden-related tweets resulted in 308,283 instances with positive sentiment (1) and 298,479 instances with negative sentiment (0).

In summary, the counts provide a breakdown of sentiment classifications (positive and negative) for both Trump and Biden tweets, as determined by the Vader and TextBlob sentiment analysis algorithms.

In order to identify the winning candidate based on the percentage of positive tweets concluded from both the data sets.

```
#percentage of ppl responded positively in the trump dataset acco
trump_ppv=(trump_p_v/(trump_p_v+trump_n_v))*100.00
#percentage of ppl responded negatively in the trump dataset acco
trump_pnv=(trump_n_v/(trump_n_v+trump_p_v))*100.00
#percentage of ppl responded positively in the Biden dataset acco
biden_ppv=(biden_p_v/(biden_p_v+biden_n_v))*100.00
#percentage of ppl responded negatively in the Biden dataset acco
biden_pnv=(biden_n_v/(biden_n_v+biden_p_v))*100.00

#percentage of ppl responded positively in the trump dataset acco
trump_pptb=(trump_p_tb/(trump_p_tb+trump_n_tb))*100.00
#percentage of ppl responded negatively in the trump dataset acco
trump_pntb=(trump_n_tb/(trump_n_tb+trump_p_tb))*100.00
#percentage of ppl responded positively in the Biden dataset acco
biden_pptb=(biden_p_tb/(biden_p_tb+biden_n_tb))*100.00
#percentage of ppl responded negatively in the Biden dataset acco
biden_pntb=(biden_n_tb/(biden_n_tb+biden_p_tb))*100.00
```

The code starts by obtaining the counts of positive and negative sentiments for Trump and Biden from both VADER and TextBlob analyses. It uses these counts to calculate the percentage of positive and negative sentiments separately for each candidate in both analyses. The code then compares the percentages of positive sentiments between Trump and Biden for both VADER and TextBlob. Based on the higher positive sentiment percentage, it determines the "winner" for each sentiment analysis method.

K. Findings and Result

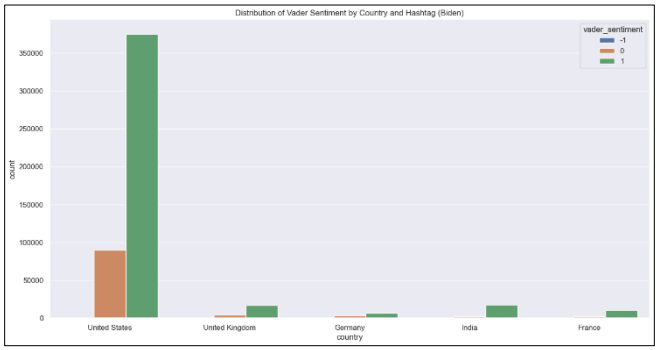


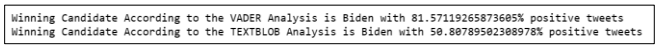
Fig: 11 distribution of Vader sentiment by country and hashtag for the hashtag #Biden

The image shows the distribution of Vader sentiment by country and hashtag for the hashtag #Biden. The countries with the highest Vader sentiment are the United States, the United Kingdom, Germany, India, and France. The United States has the highest Vader sentiment, with over 300,000 tweets. The United Kingdom has the second highest Vader sentiment, with over 250,000 tweets. Germany has the third highest Vader sentiment, with over 200,000 tweets. India has the fourth highest Vader sentiment, with over 150,000 tweets. France has the fifth highest Vader sentiment, with over 100,000 tweets.

The findings from the image suggest that there is a lot of positive sentiment towards Biden on Twitter in the United States, the United Kingdom, Germany, India, and France. This could be due to a number of factors, such as Biden's policies, his personal qualities, or the general political climate in these countries.

The same is seen across the trump dataset as well the counts were very low when compared to the bidens positive counts Hence not attaching the image.

The Result of the mentioned approach for achieving the problem statement is given in the below image:



The output indicates that, according to the sentiment analysis using VADER, Biden has a higher percentage of positive tweets (81.57%) compared to Trump. Similarly, in the TEXTBLOB analysis, Biden again emerges as the winner with 50.81% positive tweets, surpassing Trump. This suggests that, based on both VADER and TEXTBLOB analyses, Biden received a higher proportion of positive sentiments in the analyzed tweets.

L. Discussion:

- Sentiment Analysis Comparison: Sentiment score created by VADER is more accurate compared to TextBlob The sentiment calculation time increased with a larger dataset, emphasizing the need for efficient processing.
- Geographical Sentiment Variation: The sentiment analysis across different states revealed varying opinions on political participants.
- Impactful Predictor Variables: Additional predictor variables, such as user profession and age, showed potential in influencing public interest.
- Model Exploration: Random Forest model, tuned through GridSearchCV, was attempted for sentiment analysis.

BERT Transformers from Hugging Face were explored, but the extensive processing time exceeded 5 hours, making it impractical.

M. Limitations:

- Computational Resources: Processing large datasets and implementing complex models like BERT faced limitations due to hardware constraints.
- Processing Time: Text preprocessing, especially for sentiment analysis, proved time-consuming with a high volume of records.

N. Future Scope:

Data Quality Emphasis: Focus on collecting high-quality tweets for more accurate sentiment analysis.

Reducing Dataset Size: Consider reducing the dataset size while maintaining data quality to improve processing times.

Incorporating Advanced Models: Implementation of advanced models like GPT for enhanced sentiment analysis and emotion recognition.

Feature Expansion: Exploration of additional features and variables to enhance the predictive power of the models.

Parallel Processing: Utilizing parallel processing techniques to handle larger datasets and expedite computations.

Real-time Analysis: Transition towards real-time sentiment analysis to capture evolving public sentiments.

IV. REFERENCES

[1] Sayyida Tabinda Kokab ,Sohail Asghar, Shehneela Naz “Transformer-based deep learning models for the sentiment analysis of social media data”

[2] Drus Z, Khalid H. Sentiment analysis in social media and its application: Systematic literature review. Procedia Comput Sci 2019;161:707–14.

[3] E. T. Sang and J. Bos, "Predicting the 2011 Dutch senate election results with Twitter.," in Proceedings of the workshop on semantic analysis in social media, 2012.

[4] R. Upadhyay and A. Fujii, “Semantic knowledge extraction from research documents,” in Computer Science and Information Systems (FedCSIS), 2016 Federated Conference on. IEEE, 2016, pp. 439–445.

[5] Miftahul Qorib, Rahel S. Gizaw, “Impact of Sentiment Analysis for the 2020 U.S. Presidential Election on Social Media Data.

[6] Ethan Xia, Han Yue, “Tweet Sentiment Analysis of the 2020 U.S. Presidential Election”.

[7] Rao Hamza Ali, Gabriela Pinto, Evelyn Lawrie, “large-scale sentiment analysis of tweets pertaining to the 2020 US presidential election”.

[8] Quratulain Alvi , Syed Farooq Ali , Sheikh Bilal Ahmed ,” On the frontiers of Twitter data and sentiment analysis in election prediction: a review”.

[9] Hao Wang, Dogan Can, Abe Kazemzadeh, François Bar and Shrikanth Narayanan, “A System for Real-time Twitter Sentiment Analysis of 2012 U.S. Presidential Election Cycle”.

[10] Hassan Nazeer Chaudhry, Yasir Javed, “Sentiment Analysis of before and after Elections: Twitter Data of U.S. Election 2020”.

[11] Md. Rakibul Hasan1, Maisha Maliha, “Sentiment Analysis with NLP on Twitter Data”.

[12] Widodo Budiharto, Meiliana Meiliana, “Prediction and analysis of Indonesia Presidential election from Twitter using sentiment analysis”.

[13] Xiaokang Gong, Wenhao Ying, Shan Zhong, “Text Sentiment Analysis Based on Transformer and Augmentation”.

[14] Sayyida Tabinda Kokab , Sohail Asghar, “Transformer-based deep learning models for the sentiment analysis of social media data”.

[15] Team, Great Learning. “What Is Rectified Linear Unit (ReLU)? | Introduction to ReLU Activation Function.” Great Learning Blog: Free Resources What Matters to Shape Your Career!, 29 Aug. 2020,

[16] Yi, Dokkyun, et al. “An Effective Optimization Method for Machine Learning Based on ADAM.” Applied Sciences, vol. 10, no. 3, Feb. 2020, p. 1073. DOI.org (Crossref),