Prediction Of Presidential Election Results Based On Sentiment Analysis

Abstract

Social media provides a platform for individuals to articulate their opinions and sentiments on various political topics, such as elections, and thus facilitates political discussions. This study develops accurate models to predict the 2023 Nigerian elections using Twitter data. Pre-trained sentiment analysis models label tweets, and machine learning (Naive Bayes, SVM, Random Forest) predicts sentiment. Deep learning (BERT, Logistic Regression, LSTM) models analyze candidate-related tweets, with LSTM showing 88% accuracy. Insights into pre/post-election discussions are gained through word analysis and topic modelling. This research advances sentiment-based election prediction in Nigeria, highlighting LSTM's effectiveness and offering insights for analysts and policymakers. Further research is needed for real-time prediction accuracy in dynamic social media contexts. However, further research is warranted to enhance model performance, considering the dynamic nature of social media data and its implications for real-time prediction accuracy. In summary, this research offers a comprehensive exploration of predicting Nigerian elections using sentiment analysis of tweets and sheds light on the underlying topics of public discourse.

1. Introduction

The credibility of Nigeria's Independent National Electoral Commission (INEC) has been a subject of concern, as revealed by a survey by Afro Barometer where 78% of Nigerians expressed distrust in its ability to conduct credible elections, despite 71% acknowledging elections as the preferred method for choosing leaders. This sentiment gained significance in the context of Nigeria's recent general election on February 25, 2023. With nearly 40% of registered voters being youths (Egobiambu, 2023), social media emerged as a powerful platform for political campaigns. The youth, frustrated with past governments, sought change and reform, and social media became a pivotal tool to voice their opinions.

This research uses sentiment analysis and topic modelling on social media data to investigate these complex dynamics. Key questions addressed include:

1) The dominant sentiment on social media about the 2023 Nigerian Presidential election and its relation to the official results.
2) The general sentiment on social media towards the candidate contesting the election results.
3) How sentiment towards candidates and the electoral process on social media changed and the factors influencing these changes.
4) How social media sentiment matches the official results and what it means for wider societal attitudes towards the political system.

Our aim is to examine the intricate connection between social media sentiment, public perception, and the Nigerian electoral landscape, providing useful insights into the future of Nigerian politics and the electoral process.

2. Literature Review

Sentiment analysis is a computational method that analyzes large datasets, such as social media posts, to extract opinions, attitudes, and emotions for election forecasting (Liu, 2012; Pang & Lee, 2008). It belongs to polarity-based sentiment analysis, which classifies text as positive, negative, or neutral based on its emotional context (Turney, 2002). Scholars have explored various strategies for polarity-based sentiment analysis, including lexicon-based methods (Hu & Liu, 2004), machine learning algorithms (Pak & Paroubek, 2010), and deep learning techniques (Kim, 2014).

Twitter, a widely used online social networking platform, plays a crucial role in using sentiment analysis for election predictions. With more than 300 million users, it has become a significant platform for politicians, including presidents, to share news and posts, communicate with the population, and hopefully sway votes in their favour (Dixon, 2022). In 2019, following an in-depth analysis, The New York Times reported that Mr Donald Trump, who was the president of the United States at the time, was an avid Twitter user (Shear et al., 2019). “Twitter was a political tool that had helped get him elected” (Shear et al., 2019).

Lexicon-based approaches have gained popularity because of their simplicity and effectiveness, as noted by Mohammad and Turney (2010). Nevertheless, recent progress in machine learning and deep learning methods has demonstrated significant potential in comprehending the intricacies of language and context, as emphasized by Poria et al. (2018).

Sharma & Moh (2016) used sentiment analysis on Hindi tweets to predict the Indian general elections in 2016. They use three techniques to classify tweets as positive, negative or neutral and find that Support Vector Machine is the best, with 78.4% accuracy. They predict that Bhartiya Janta Party (BJP) will win more elections, which is consistent with the actual result.

Oyewola et al. (2023) apply sentiment analysis to tweets pertaining to the 2023 Nigerian presidential election. They employ three models to categorize tweets towards the candidates as positive, neutral, or negative and discover that two-stage residual long short-term memory (TSRLSTM) performs the best. They examine the implications of their findings for political science research and practice. Moreover, Barghuthi and Said (2020) utilize sentiment analysis on tweets to make accurate forecasts for Turkey’s 2018 presidential election, which correspond with the actual outcomes.

3. Methodology

The project will use the following pipeline:

- Data Collection
- Data Preprocessing
- Labelling the dataset: TextBlob, VADER
- Sentiment Analysis: Naive Bayes, Support Vector Machine (SVM), and Random Forest
- Election Result Prediction: Using Logistic Regression, BERT, and LSTM
- Model Evaluation and Results
- Analysis of critical issues
- Interpretation and results

3.1 Data Collection

This study uses unlabeled datasets from Kaggle, consisting of approximately 22,000 tweets per candidate and political party. The tweets were collected during the peak election period from December 21, 2022, to February 21, 2023. A dataset of tweets after the election was also gathered from Kaggle for further analysis.

3.2 Data Cleaning and Preparation

Data cleaning ensures that the data used for analysis or modelling is accurate, consistent, and reliable, leading to more meaningful and accurate insights and decision-making.

A column indicating the data frame name was added to each dataset for easier access to each data frame through filtering. Furthermore, a function, *add_candidate_column*, was defined to associate a specific candidate name with each row of the data frame.

The *pd.concat()* function was evoked to merge the multiple data frames into a single, consolidated dataset, *combined_df,* for further analysis.

The following pre-processing techniques were undertaken to enhance the quality of the data:

a. Drop unnecessary columns and duplicates.
b. Expand contractions in the text.
c. Remove stopwords from the text.
d. Strip numeric values and punctuation from the text.
e. Remove extra spaces, greetings, and other unimportant words.
f. Remove mentions, URLs, and hashtags from the text.
g. Convert all text to lowercase.

3.3 Labelling Dataset

Manual labelling of a large dataset can be time-consuming and resource-intensive. In this report, we suggest using established sentiment analysis tools: Valence Aware Dictionary and Sentiment Reasoner (VADER), the Natural Language Toolkit (NLTK) sentiment analyzer, and TextBlob to automatically label the dataset, enabling the development of an optimal sentiment analysis model.

The dataset is split into two equal parts: one for training and testing the model and the other for making predictions with the model. The study uses VADER and TextBlob to generate sentiment labels that match human intuition, reducing manual effort and time for labelling the dataset. After removing unnecessary columns, the data is divided into two data frames, one for labelling sentiment and the other for assessing the trained sentiment analysis model.

3.4 Sentiment Labelling Results

The following charts show the results obtained from labelling the dataset using TextBlob and VADER:

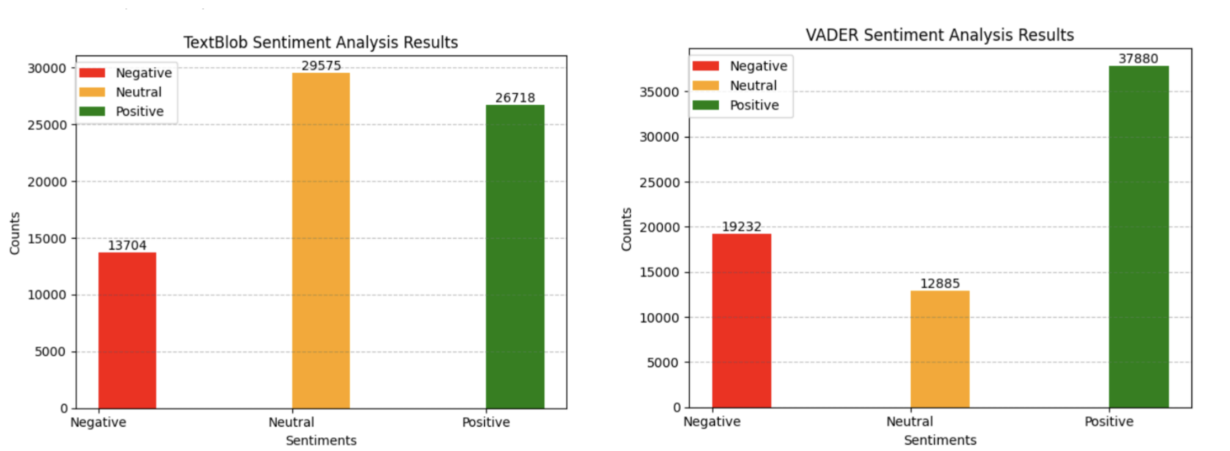

Fig. 1: TextBlob and VADER Sentiment Analysis Results

The disparity in the results between VADER and TextBlob arises due to the distinct algorithms and language processing they employ. VADER is attuned to social media language and expressions, while TextBlob employs a broader machine learning model. Differences in handling informal language, context, lexicons, and sensitivity to nuanced sentiment contribute to the variations. We will make use of both datasets to maintain integrity.

4. Sentiment Analysis Algorithms

4.1 Multinomial Naive Bayes Classifier

The multinomial Naive Bayes classifier is a probabilistic machine learning model designed to improve the classic Naive Bayes algorithm to handle multiple classes and discrete feature spaces, making it especially suitable for categorizing text and other categorical data issues.

I. Transforming Text Data

We used *CountVectorizer* from the scikit-learn library to convert text data into a matrix of token counts. We then divided the data into training and testing subsets. The algorithm learns the conditional probability of each feature in each class during training. Then, it uses Bayes' theorem to compute the probability of each class given the observed features during prediction. The class with the highest probability is chosen as the predicted class.

II. Splitting the Dataset

The *train_test_split* function from scikit-learn is evoked to divide the dataset into two subsets: one for training the model (80%) and the other for assessing its performance (20%). The *split_data* function is introduced to encapsulate this splitting procedure, taking input matrices of token counts (or features) and corresponding sentiment labels.

III. Compiling, Fitting, and Evaluating the Model

The classifier is created using the MultinomialNB class from the naive_bayes module under the scikit-learn library. The fit method is then called on the classifier object, passing the training data (X_train) and corresponding target sentiment labels (Y_train) as arguments.

IV. Unigrams, Bigrams and Trigrams

Unigrams, bigrams, and trigrams represent fundamental n-gram models that capture varying degrees of linguistic context within a sequence of words. These n-gram models help capture contextual information that contributes to the sentiment expressed in a sentence or document. Evaluating multiple settings helps in selecting the most suitable configuration for sentiment analysis tasks. The tables below summarize the n-gram results:

UNIGRAMS						
VADER				TEXTBLOB		
	Precision	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**
Positive	0.85	0.90	0.87	0.85	0.93	0.89
Negative	0.78	0.69	0.73	0.83	0.69	0.75
Accuracy: 82.74				**Accuracy:** 84.58		

Table 1: Performance of Naive Bayes models using Unigram settings

BIGRAMS						
VADER				TEXTBLOB		
	Precision	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**
Positive	0.80	0.75	0.77	0.81	0.70	0.75
Negative	0.56	0.63	0.59	0.54	0.67	0.60
Accuracy: 70.70				**Accuracy:** 69.16		

Table 2: Performance of Naive Bayes models using Bigram settings

TRIGRAMS						
VADER				TEXTBLOB		
	Precision	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**
Positive	0.79	0.44	0.57	0.79	0.34	0.48
Negative	0.42	0.77	0.54	0.39	0.83	0.53
Accuracy: 55.41				**Accuracy:** 50.60		

Table 3: Performance of Naive Bayes models using Trigram settings

As a result of the impressive evaluation metrics, it was concluded that using unigrams makes the best predictor, and therefore, other algorithms would be tested using unigram settings.

4.2 Support Vector Machine (SVM)

SVM is a sophisticated supervised learning algorithm with broad applications in classification and regression tasks. It finds the optimal hyperplane to separate different classes in the input data space, maximizing the margin and performing well even with limited training samples and features. The models achieve high accuracy and generally good performance across all evaluation metrics.

I. Hyperparameter Tuning

Having split the dataset into 80% for training and 20% for testing, this process involves creating a classifier pipeline with SVM and TF-IDF vectorization, and then performing a grid search to tune hyperparameters for sentiment analysis on the VADER and TextBlob datasets.

The *perform_sentiment_analysis_grid_search* function is used to conduct the grid search, finding the best hyperparameter configuration and optimizing the SVM model's performance on the datasets.

II. Model Training

We define a function capable of taking several parameters, called, *train_sentiment_analysis_SVM_model,* that is responsible for training. The function initializes a TF-IDF vectorizer with specified configurations and fits it to the entire dataset; transforms training and testing data into TF-IDF features; initializes an SVM classifier with defined parameters and trains it on transformed training data; predicts labels for testing data using the trained SVM model, yielding svm_predictions; and ultimately returns both the trained SVM model and predictions.

III. Model Evaluation

The *print_sentiment_analysis_results* function provides a comprehensive summary of the sentiment analysis results. The following table summarises the results obtained:

SVM						
	VADER			TEXTBLOB		
	Precision	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**
Positive	0.90	0.93	0.92	0.94	0.96	0.95
Negative	0.86	0.81	0.83	0.93	0.89	0.91

Accuracy: 89.01	**Accuracy:** 93.89

Table 4: Performance of SVM models

4.3 Random Forest Classifier

The Random Forest classifier improves the accuracy of sentiment prediction by using an ensemble learning technique that merges many decision trees and uses different sentiment indicators from text data.

I. Model Preparations

Like the previous models, the dataset for the RF Classifier is split, and then we create the TF-IDF vectorized data.

II. Model Training

The function *train_random_forest* takes TF-IDF vectorized training data (X_train), training labels (y_train), and TF-IDF vectorizer testing data (X_test) as input. It initialises a Random Forest classifier (RF) with specified parameters and trains it using the training data and labels. Then, it predicts labels for the testing data using the trained random forest model and returns the predicted labels (y_pred).

III. Model Evaluation

The models achieve moderate to high accuracy and demonstrate varying performance across the evaluation metrics. The following table summarises the results obtained:

Random Forest Classifier						
	VADER			**TEXTBLOB**		
	Precision	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**
Positive	0.82	0.95	0.88	0.89	0.96	0.93
Negative	0.86	0.59	0.70	0.92	0.77	0.8422
Accuracy: 83.03				**Accuracy:** 89.98		

Table 5: Performance of Random Forest models

Based on the classification reports generated, the SVM model using the TextBlob dataset achieved the highest accuracy of 93.89%, followed closely by the Random Forest model using the TextBlob dataset with an accuracy of 89.98%. This made SVM the best model implementation to predict the sentiments on the deployed dataframe.

5. Election Results Prediction Model

The assumption is that sentiment in the tweets directly relates to voting behaviour. Positive expressions indicate an intention to vote for a candidate, while negative sentiment suggests voting against them. However, the model acknowledges potential biases in sentiment analysis due to the assumption that social media represents the entire voting population without demographic over- or under-representation on different platforms.

5.1 Logistic Regression

The model estimates the probability that an instance belongs to a specific class (often represented as 1) and its complementary class (typically represented as 0).

I. Model Preparation and Training

In the process, sentiment labels 'Positive' and 'Negative' are mapped to their corresponding numerical values, 1 and 0, respectively. This mapping function is then applied to separate DataFrame columns in vader_data and textblob_data. The resulting numerical values are stored in new columns named 'VADER_Sentiment' and 'TextBlob_Sentiment,' respectively. This conversion facilitates the suitability of the sentiment labels for model training and evaluation. We then create a function that splits the data into 80% training and 20% testing for both positive and negative samples while ensuring class balance. Another function called *tokenize_data* tokenizes text data using the Keras tokenizer. We create a function that trains the model by first fitting a logistic regression model with cross-validation and then applying Recursive Feature Elimination RFE to select the specified number of features. Next, it creates a new dataset with the selected features and fits a Random Forest classifier using these features. The function returns the trained ensemble model.

II. Model Evaluation

The function *predict_and_evaluate_model* prints the classification report, which includes precision, recall, F1-score, and support metrics, comparing the binary predictions to the true labels. The following table summarizes the results achieved:

Logistic Regression						
VADER				TEXTBLOB		
	Precision	Recall	F1 Score	Precision	Recall	F1 Score
0	0.50	0.22	0.31	0.55	0.15	0.23
1	0.62	0.85	0.72	0.65	0.93	0.76
Accuracy: 60				**Accuracy:** 64		

Table 6: Performance of Logistic Regression models

5.2 Bidirectional Encoder Representations from Transformers (BERT)

BERT is a state-of-the-art transformer-based neural network architecture renowned for its contextualized language understanding. We used the *bert-base-uncased* model, which is a variant of BERT that has been pre-trained on a large corpus of uncased English text. The tokenizer is responsible for converting text into tokens that can be processed by the model.

I. Balancing the Sentiment Labels

We balance the dataset by sampling and preprocessing the data on the VADER and TextBlob sentiment analysis datasets, creating balanced datasets by selecting an equal number of positive and negative tweets from each and shuffling them for training and evaluation purposes.

II. Splitting the Data for Train and Test

The *create_datasets* function divides input text data with sentiment labels into stratified training, validation, and test subsets, forming dictionaries for datasets and labels. It produces separate sets for VADER and TextBlob data

III. Data Tokenization

The tokenization function utilizes the BERT tokenizer to tokenize input data, allowing for BERT-based tokenization of textual data. It returns TensorFlow tensors of tokenized sequences with options for padding, truncation, and maximum sequence length.

IV. Base BERT Model

We create a binary classification model using a pre-trained BERT transformer. It defines input layers for the BERT model with a specified sequence length. The model is compiled using the Adam optimizer with a learning rate of 1e-5, binary cross-entropy loss, and accuracy as the evaluation metric.

V. Evaluating BERT Base Model

The *test_result* function evaluates a binary classification model on a test dataset by applying the BERT tokenizer to the text data. It then makes predictions and converts probabilities into binary predictions based on class-specific thresholds derived from the training data distribution.

VI. Applying Hyperparameters

The *train_model_on_subset* function trains a compiled Keras binary classification model on a reduced portion of training data. It involves tokenizing the subset and validation data through the BERT tokenizer and then training the model with specified epochs and batch size. Post-training, the function makes predictions on the validation set and optimizes the binary prediction threshold using the Receiver Operating Characteristic (ROC) curve.

VII. Evaluation Metrics

This takes a trained Keras binary classification model, true labels, and predicted labels for the validation set and plots the confusion matrix using a heatmap with annotations. A summary of the results is as follows:

BERT						
	VADER			TEXTBLOB		
	Precision	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**
0	0.54	0.61	0.57	0.55	0.62	0.58
1	0.55	0.48	0.51	0.57	0.49	0.52
Accuracy: 0.54				**Accuracy:** 0.56		

Table 7: Performance of BERT models

5.3 Long Short-Term Memory (LSTM)

LSTM is a variant of RNN that handles sequence data and overcomes the vanishing gradient problem. It employs memory cells and gates to store and retrieve information over extended periods. The gates determine what to store, remove, and output. This enables LSTM to capture context and excel in tasks such as translation, sentiment analysis, forecasting, and speech recognition.

I. Tokenizing Data

This function handles text data by tokenizing it, converting it to sequences, padding the sequences to a fixed length, and one-hot encoding the target sentiment values.

II. Initializing the Model and Defining Hyperparameters

The function prints a summary of the model's architecture, and it is then called twice with different input lengths (one for each dataset - VADER and TextBlob) to create separate LSTM models for each dataset.

III. Fit and Train the Model

The function calculates class weights based on the bias in the target data and fits the model on the training data while validating the test data.

IV. Evaluating the Model

The function extracts evaluation metrics from fitted LSTM models for VADER and TextBlob data and defines a function *evaluate_model* to evaluate and print the metrics using test data, including confusion matrix visualization through a heatmap, called twice for each model. Here is a summary of the results obtained:

LSTM						
VADER				**TEXTBLOB**		
	Precision	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**
0	0.92	0.83	0.87	0.90	0.85	0.88
1	0.84	0.93	0.88	0.86	0.91	0.88
Accuracy: 0.88				**Accuracy:** 0.88		

Table 8: Performance of the LSTM model

V. Visualizing Training and Validation Curves

The function defines a function, *plot_accuracy,* that creates line plots for training and validation accuracy and loss curves based on provided data and sentiment model names, and it is called twice to display the curves for the VADER data LSTM model and the TextBlob data LSTM model.

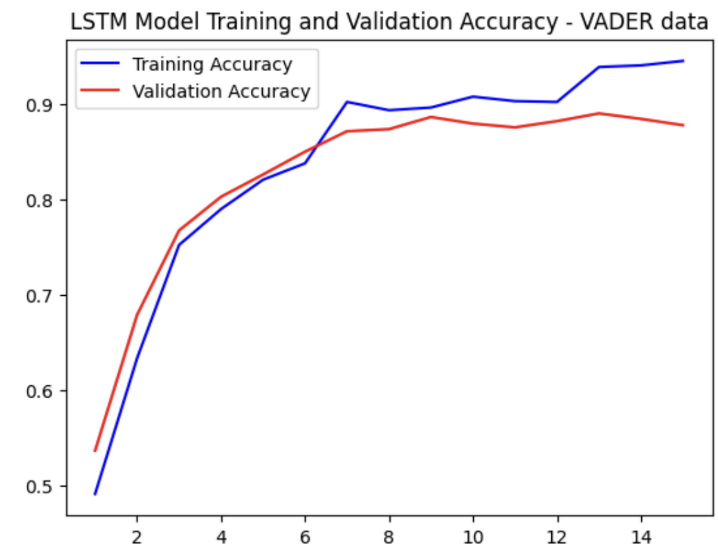

Fig. 2: VADER LSTM Validation Accuracy

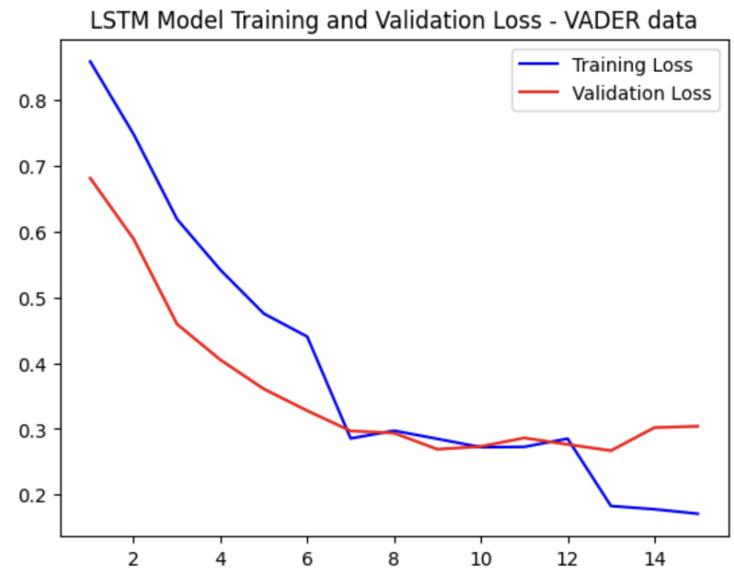

Fig. 3: VADER LSTM Validation Loss

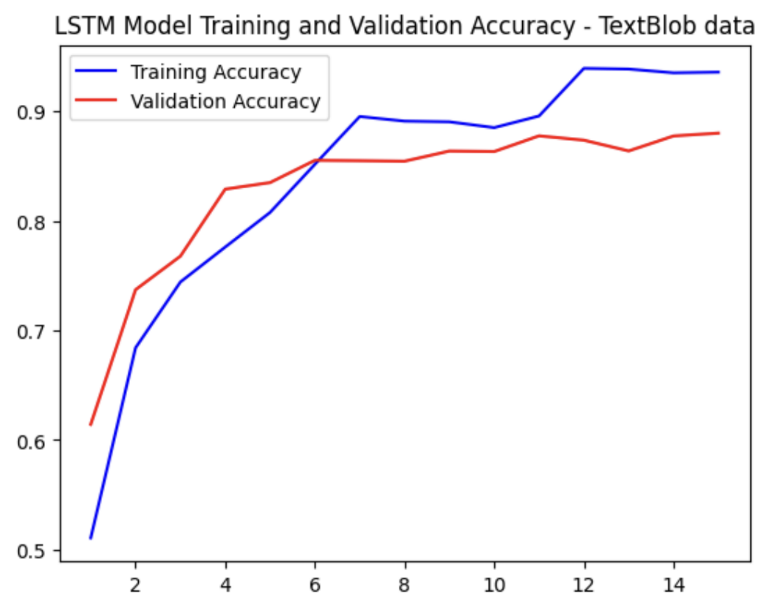

Fig. 4: TextBlob LSTM Validation Accuracy

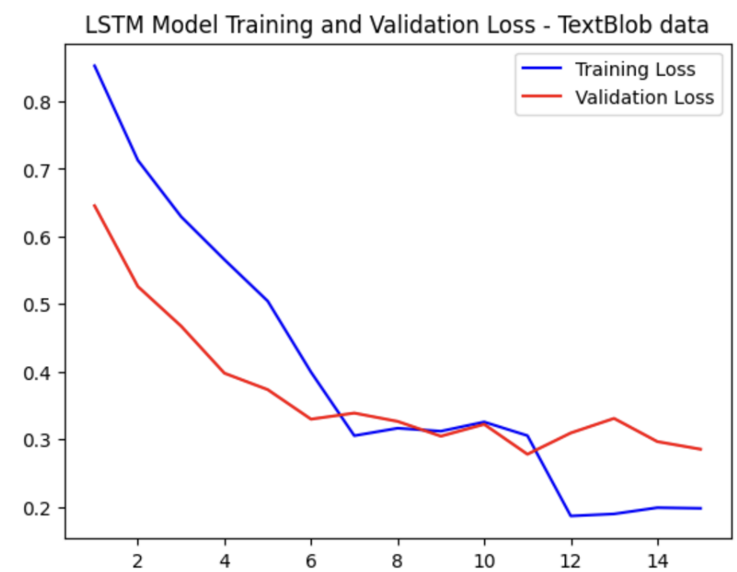

Fig. 5: TextBlob LSTM Validation Loss

With an accuracy score of 88%, the LSTM model shows the best performance in comparison to the Logistic Regression and BERT models in classifying the sentiments expressed in social media posts about the 2023 Nigerian Presidential election, Thus, making it the top-performing model for election prediction in this specific scenario.

6. Election Prediction Using LSTM

We proceed to forecast the labels of the tweets within the validation data, which we refer to as *final_prediction*, followed by the generation of a grouped bar plot. Subsequently, a comparative analysis will be conducted with real-life scenarios, facilitating the identification and selection of the most effective sentiment pre-trained model. The following steps were taken to achieve this:

1. Loading the validation data
2. Balancing the number of tweets per candidate to validate our model
3. Preprocess the data
4. Make predictions
5. Convert probabilities to binary predictions - this is to compare the sentiment predictions from both VADER and TextBlob methods and analyze how well they perform in classifying sentiments in tweets related to the 2023 Nigerian Presidential election.
6. Convert the binary to yes or no
7. Bar plot to choose the best pre-trained sentiment analysis models

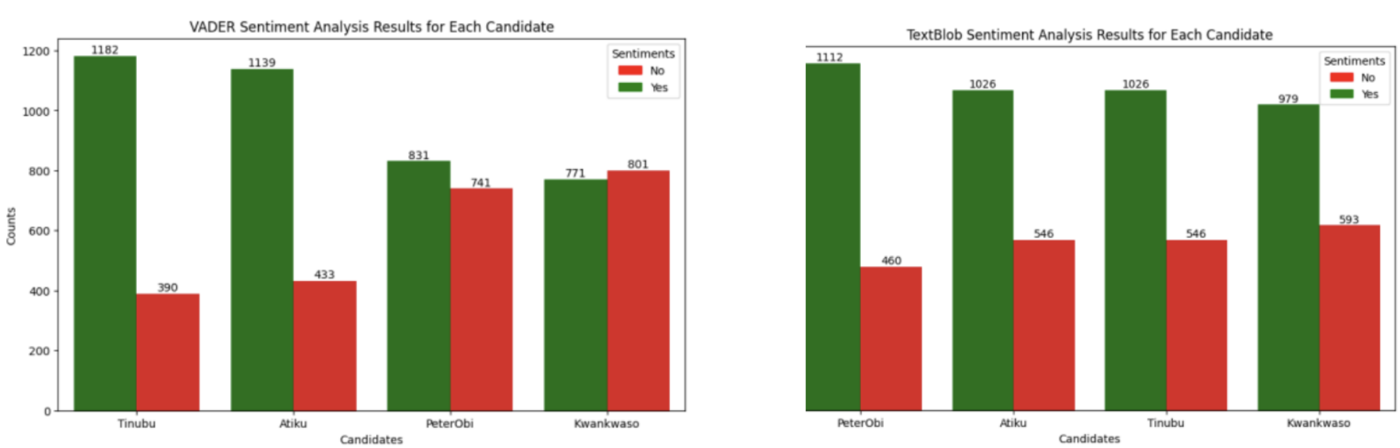

Fig. 6: VADER and TextBlob Sentiment Analysis Results

Based on the election results presented by Stears Co., the outcome showed Tinubu Bola Ahmed obtained the highest number of votes (8,794,726 votes, 36.61%), followed by Abubakar Atiku

(6,984,520 votes, 29.07%), Obi Peter Gregory (6,101,533 votes, 25.40%), and Musa Mohammed Rabiu Kwankwaso (1,496,687 votes, 6.23%). The VADER data predictions were deemed accurate based on the provided plots, leading to the decision to drop the TextBlob data. Consequently, the VADER model was identified as the most suitable pre-trained model for labelling tweets.

8. Bar plot for yes and no votes for all candidates

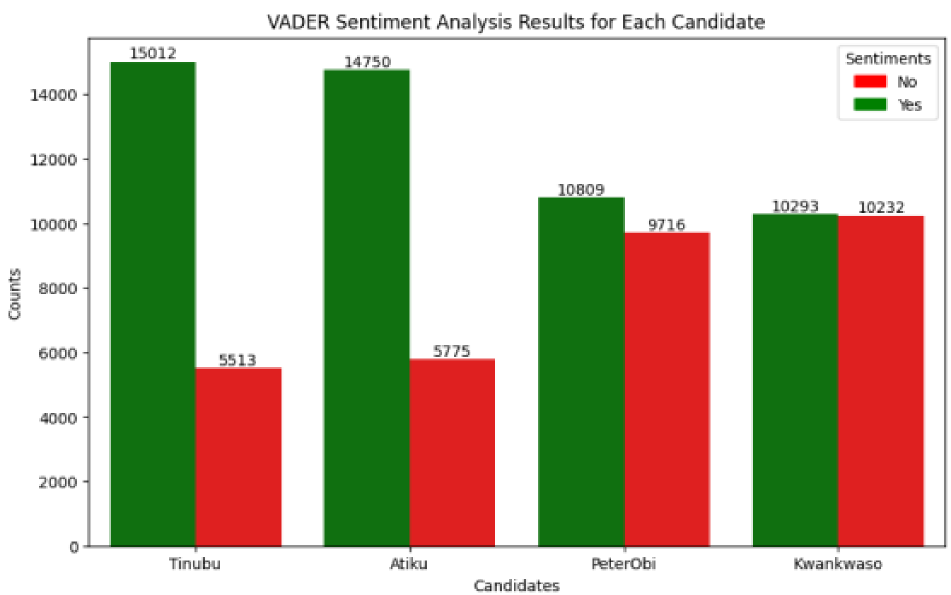

Fig. 7: VADER Sentiment Analysis Results

7. Results and Discussions

We can observe the performance of different sentiment analysis methods and tools, such as lexicon-based approaches (VADER and TextBlob) and machine learning algorithms (Random Forest, SVM, and Naive Bayes), in detecting and classifying sentiment in tweets about the 2023 Nigerian Presidential election.

The Random Forest Model using VADER achieved an accuracy of 83.06%; using TextBlob, it achieved a higher accuracy of 89.87%. The SVM Model using VADER achieved an accuracy of 89.01% and performed even better using TextBlob, with an accuracy of 93.89%. The Naive Bayes Model using VADER had the lowest accuracy of 43.23% but achieved a slightly higher accuracy of 45.75% using TextBlob.

All models showed varying levels of precision, recall, and F1-score for both negative and positive sentiments. The SVM models generally had higher precision, recall, and F1-scores compared to other models, indicating better performance in detecting sentiments accurately. The machine learning algorithms (Random Forest, SVM, and Naive Bayes) demonstrated relatively better performance.

The literature review underscores the significance of sentiment analysis in this context, with a focus on polarity-based approaches and the potential of machine learning and deep learning methods.

The evaluation metrics for Logistic Regression, BERT, and LSTM models using VADER and TextBlob data were considered. The LSTM model outperformed the others consistently, achieving F1-scores of 0.88 for both datasets. In contrast, BERT demonstrated relatively lower performance, and Logistic Regression fell in between.

The sentiment analysis during the 2023 Nigerian Presidential election indicated a prevailing positive sentiment on social media towards all candidates. Tinubu, Atiku, and Peter Obi received higher positive sentiments, while Kwankwaso's sentiment was more balanced. Notably, Atiku, who contested the election, garnered the second-highest number of positive sentiments. The sentiments expressed on social media aligned remarkably with the official election results, highlighting the significance of social media in reflecting public sentiment and its impact on electoral outcomes.

These findings align with the literature, highlighting the relevance of advanced deep learning methods like LSTM. Lexicon-based approaches, though effective, did not match the performance of machine learning and deep learning models. Twitter's role as a significant platform for political discussions was emphasized, supporting the use of social media data for sentiment analysis in election predictions.

Topic Modelling using LDA

LDA is a statistical technique that can discover topics in a collection of text. LDA assumes that each text in the collection has several topics and that each topic has a probability of using certain words. LDA aims to find the probabilities of topics for each text and the probabilities of words for each topic.

From pre-election tweets, the model extracted topics pertaining to political figures, states, candidates, support, opinions, voting, religion, and law. From post-election tweets, the model identified topics related to alliances, economy, law, outcomes, gratitude, and news.

Content Analysis

The word clouds of tweets before and after the elections show different themes and topics of public discourse. Before the elections, tweets express support, excitement, and concern for candidates and issues. After the elections, tweets focus on legal and electoral matters, such as fraud, charges, and disputes.

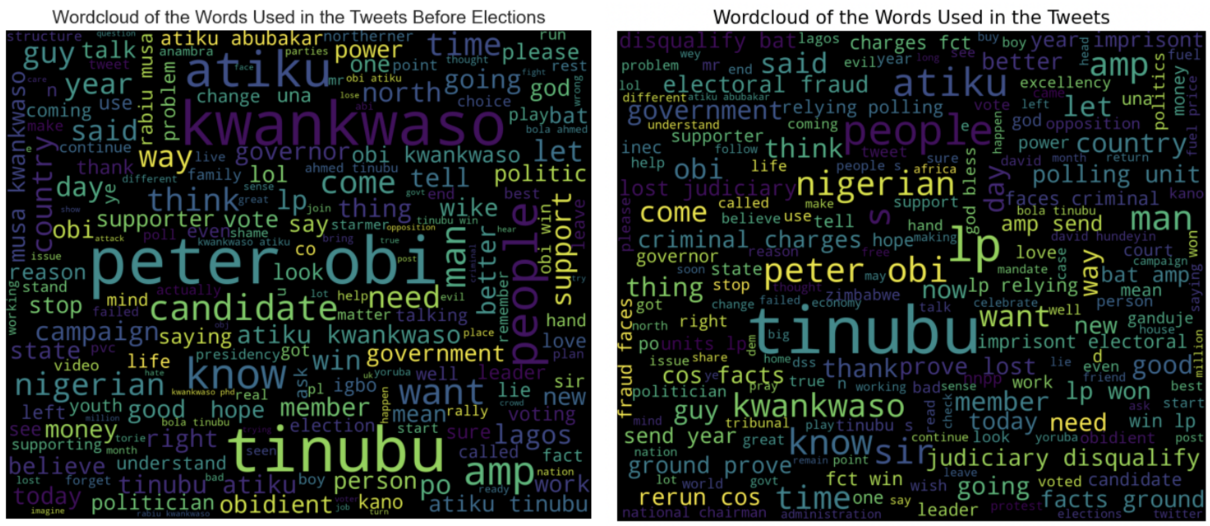

Fig. 8: WordCloud of Tweets Before and After Elections

The word frequencies of tweets before and after the elections indicate the common topics and keywords on social media during those periods. Before the elections, tweets mention candidates, regions, campaigns, and elections. After the elections, tweets discuss political figures, government, legal issues, protests, affiliation, social issues, gratitude, and location-specific topics.

Conclusion

This study demonstrates the feasibility and utility of using sentiment analysis of Twitter data to predict the 2023 Nigerian elections. By applying pre-trained and deep learning models to a large corpus of tweets, the study achieves high accuracy in classifying the sentiments and preferences of the Nigerian electorate. The study also reveals the main themes and issues that shape public opinion and discourse on the candidates and the election process. The findings of this study contribute to the growing literature on sentiment-based election prediction, especially in the context of Nigeria, where social media plays a vital role in political communication. The study also showcases the potential of LSTM as a powerful deep learning technique for sentiment analysis. However, the study acknowledges the limitations and challenges of using social media data for real-time prediction, as it is subject to rapid changes and fluctuations. Therefore, the study suggests directions for future research to improve the robustness and reliability of the models and to incorporate more features and variables that can capture the complexity and diversity of social media sentiment.

Recommendations

As we progress with the project, it is recommended to focus on ensuring the data collection process is comprehensive and unbiased. Diverse sources of information should be considered to obtain a well-rounded view of the electorate's sentiments. Moreover, it is essential to continuously validate and fine-tune the machine learning model to improve its accuracy and reliability in predicting election outcomes. By addressing these aspects with diligence and attention, we can maximize the project's impact and make informed contributions to the understanding of Nigeria's political landscape.

References

1. AfricaNews (2023) Nigeria’s opposition petitions to overturn election result. Africanews. Available online: https://www.africanews.com/2023/03/22/nigerias-opposition-petitions-to-overturn-election-result [Accessed 11 Aug. 2023].
2. Afrobarometer (2023) Nigerians want competitive elections but don’t trust the electoral commission. Available online: https://www.afrobarometer.org/wp-content/uploads/2023/02/R9-News-release-Nigerians-want-comp etitiveelections-but-dont-trust-INEC-Afrobarometer-2feb23.pdf [Accessed 27/03/2023].
3. Afrobarometer. (2021). Afrobarometer Round 8: The Quality of Democracy and Governance in Africa (Survey No.R8).[Data file and codebook]. Retrieved from http://www.afrobarometer.org [Accessed 27/03/2023].
4. Ahmed, K., Rahman, S. M. M., Amin, M. R., & Afrin, Z. (2021). Predicting US Presidential Election Outcomes with Natural Language Processing and Machine Learning Techniques. IEEE Access, 9, 43848-43859.
5. Arise News (2023) We Won the Election and We Will Prove it to Nigerians - Peter Obi [Video]. Available online: https://www.youtube.com/watch?v=mX9y_spUTQs&t=14s [Accessed 28/03/2023]
6. Barghuthi, N. & Said, H. (2020) Sentiment Analysis on Predicting Presidential Election: Twitter Used Case, 105–117. Available online: https://www.researchgate.net/publication/339844280_Sentiment_Analysis_on_Predicting_Presidential_Election_Twitter_Used_Case [Accessed 28/03/23]

7. Bot, M.D. (2022) Bureaucracy, Independent National Electoral Commission (INEC) and the electoral process in Nigeria. International Journal of Intellectual Discourse, [online] 5(3), 136–142. Available online: https://ijidjournal.org/index.php/ijid/article/view/288/205 [Accessed 6 Aug. 2023].
8. Dixon, S.J. (2022) Social media and politics in the United States. Statista. Available online: https://www.statista.com/topics/3723/social-media-and-politics-in-the-united-states/#topicOverview.
9. Egobiambu, E. (2023) 2023 Elections Belong To Young People, Says INEC . Channels TV. Available online: https://www.channelstv.com/2023/01/17/2023-elections-belong-to-young-people-says-inec/ [Accessed 11 Aug. 2023].
10. Ejekwonyilo, A. (2023) #NigeriaDecides2023: Atiku challenges Tinubu's victory in court, raises five grounds. Premium Times Nigeria. Available online: https://www.premiumtimesng.com/news/589573-nigeriadecides2023-atiku-challenges-tinubus-victory-in-court-raises-five-grounds.html [Accessed 3 Apr. 2023].
11. Ejekwonyilo, Ameh (2023). #NigeriaDecides2023: Atiku Challenges Tinubu's Victory in Court, Raises Five Grounds. Premium Times Nigeria, www.premiumtimesng.com/news/589573-nigeriadecides2023-atiku-challenges-tinubus-victory-in-court-raises-five-grounds.html. Accessed 3 Apr. 2023.
12. Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 168-177.
13. Kim, Y. (2014). Convolutional neural networks for sentence classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1746-1751.
14. Liu, B. (2012). Sentiment analysis and opinion mining. Synthesis Lectures on Human Language Technologies, 5(1), 1-167.
15. Mohammad, S. & Turney, P. (2010-06) Emotions evoked by common words and phrases: Using mechanical Turk to create an emotion lexicon. Los Angeles, CA: Association for Computational Linguistics.

16. Nigeria's Obi petitions court over disputed presidential election. Available online: Nigeria's Obi petitions court over disputed presidential election [Accessed 28/03/2023].
17. Oyewola, D. O., Oladimeji, L. A., Julius, S. O., Kachalla, L. B. & Dada, E. G. (2023) Optimizing sentiment analysis of Nigerian 2023 presidential election using two-stage residual long short term memory. Heliyon, 9(4), e14836–e14836.
18. Pak, A., & Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. Proceedings of the 7th conference on International Language Resources and Evaluation, 1320-1326.
19. Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, 2(1–2), 1–135.
20. Poria, S., Hussain, A. and Cambria, E. (2018) Multimodal Sentiment Analysis. Socio-Affective Computing. Cham: Springer International Publishing.
21. Presidential election results and data 2023 - Stears Elections. (n.d.) www.stears.co. Available online: https://www.stears.co/elections/2023/president/.
22. Presidential Election Results Using Natural Language Processing Techniques. Journal of Computer Science and Its Applications, 25(1), 31-43.
23. Sharma, P. and Moh, T.-S. (2016) Prediction of Indian election using sentiment analysis on Hindi Twitter - IEEE Conference Publication. Ieee.org. Available online: https://ieeexplore.ieee.org/document/7840818. [Accessed: 3 Apr. 2023]
24. Shear, M.D., Haberman, M., Confessore, N., Yourish, K., Buchanan, L. and Collins, K. (2019) How Trump Reshaped the Presidency in Over 11,000 Tweets. The New York Times. [online] 2 Nov. Available online: https://www.nytimes.com/interactive/2019/11/02/us/politics/trump-twitter-presidency.html [Accessed 10 Aug. 2023].
25. Stears' poll predicts who will be Nigeria's next president. (2023) www.stears.co. Available online: https://www.stears.co/premium/article/stears-poll-predicts-nigerias-next-president/. [Accessed 23 June 2023]
26. Turney, P. D. (2002). Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, 417-424.