CLASSIFICATION OF POLITICAL OPINIONS : AN EXPLAINABLE AI APPROACH FOR SINHALA SENTIMENT ANALYSIS

A PROJECT REPORT PRESENTED BY

AYESHA CHANDRASENA

to the

DEPARTMENT OF STATISTICS AND COMPUTER SCIENCE

in partial fulfillment of the requirement
for the award of the degree of

BSc.(Hons) in Data Science

of the

UNIVERSITY OF PERADENIYA
SRI LANKA
2023

ABSTRACT

In the era of digital media, political opinions expressed on social media platforms have become a rich source of insights into public sentiment. Is it possible to effectively model political sentiment to reflect a nation's voting intentions during an election campaign? This research focuses on classifying political opinions in Sinhala, using Explainable AI techniques for sentiment analysis. The primary objective of this research is to develop an explainable AI model to predict the election results for the presidential election of 2024 in Sri Lanka by analyzing sentiments and opinions in social media. We used the recent Sri Lankan presidential election as a case study to investigate the potential of modeling political sentiment through the mining of social networks. The study compares advanced AI models such as BERT (sinBERT and AshenBERTo) and supervised machine learning techniques for sentiment analysis in the Sinhala language. Furthermore, the research highlights the potential of AI-driven sentiment analysis as a predictive tool for election outcomes. The results show that BERT based models (AshenBERTo and SinBERT) perform better than machine learning models in Sinhala text classification. One of our BERT based political comment classification model obtained a 94% F1-score with high precision and recall. Our sentiment classification model obtained a 77% F1-score with high precision and recall values. Furthermore, fine-tuned sinBERT model obtained high accuracy of 0.95 with a precision of 0.90 and recall of 0.95 in classifying among political parties. The proposed approach demonstrates high accuracy in classifying political opinions and reveals significant patterns and trends in public sentiment. Furthermore, the proposed method was able to accurately predict the winning party, first runner-up and second runner-up in the presidential election, of 2024. This work contributes to the growing field of Sinhala NLP, offering a framework for the intersection of AI and political data analysis.

Keywords : Sentiment Analysis, BERT for classification, Sri Lankan Politics, Machine Learning, Explainable AI

DECLARATION

I do hereby declare that the work reported in this research thesis was exclusively carried out by me under the supervision of Dr. Hemalika T. K. Abeysundara and Dr. Erunika O. Dayaratna. It describes the results of my own independent word except where due reference has been made in the text. No part of this research thesis has been submitted earlier or concurrently for the same or any other degree.

Date: .....

.....
AYESHA CHANDRASENA

Certified by:

1. Supervisor:

Date: .....

Signature: .....

2. Head of the Department: name of the HOD

Date: .....

Signature: .....

ACKNOWLEDGEMENT

First and foremost, I wish to express my sincere gratitude to my supervisors Dr. Erunika O. Dayaratna, Department of Statistics and Computer Science, Faculty of Science, University of Peradeniya and Dr. Hemalika T. K. Abeysundara, Department of Statistics and Computer Science, Faculty of Science, University of Peradeniya, who were abundantly helpful and offered invaluable assistance, support and guidance to make this research project a success. Their expertise, insightful feedback and encouragement have been essential in shaping the quality of my work.

Beside my supervisors, I would like to express my gratitude to Dr. Sachith Abeysundara, Head of the Department of Statistics and Computer Science, Faculty of Science, University of Peradeniya and Dr. Jagath Senarathne, Department of Statistics and Computer Science, Faculty of Science, University of Peradeniya for arranging this research opportunity.

Words are inadequate in offering my thanks to all the lecturers and temporary academic staff members of Department of Statistics and Computer Science, Faculty of Science, University of Peradeniya. Finally, I would like to express my heartfelt gratitude to my beloved parents and family for their blessings, my dear colleagues for their support and wishes towards the successful completion of my research project.

Contents

List of Tables

List of Figures

CHAPTER 1

INTRODUCTION

1.1 Background of the Study

The 2024 presidential election was the first election in which people were given the opportunity to go to polls since the nationwide protest movement (Aragalaya) in 2022, which spurred the resignation of former president Gotabaya Rajapaksha. Following former president Gotabaya Rajapaksa's resignation in July 2022, Ranil Wickremesinghe was indirectly elected by a parliamentary vote on July 20, 2022, to complete the former president's term. In this context, the future of the economy and the reduction of the cost of living remain key campaign themes. Many see the election as a choice between the continuation of existing policies and leaders and the introduction of new leadership and approaches.

Namal Rajapaksa, the eldest son of former president Mahinda Rajapaksa (who served from 2005 to 2015), was the youngest candidate. He represented the Sri Lanka Podujana Peramuna, a breakaway faction of the Sri Lanka Freedom Party. President Wickremesinghe, who entered Parliament through the United National Party (UNP), is contesting as an independent candidate. Most of the other major candidates represent political coalitions. Sajith Premadasa, the son of former president Ranasinghe Premadasa, is the leader of the opposition and was the runner-up in the last presidential election. He is contesting under Samagi Jana Balawegaya, a breakaway coalition of the UNP, the party represented by his father. Anura Kumara Dissanayake of the National People's Power has gained ground over the past year as an alternate political force. He is the leader of Janatha Vimukthi Peramuna (People's Liberation Front), a Marxist-Leninist party that led two revolutionary movements in Sri Lanka in the 1970s and 1980s. Governed by articles 88 and 89 of the Constitution, any citizen of Sri Lanka who is 18 years of age or older and is correctly registered in the electoral register is eligible to vote for an election in Sri Lanka. As the latest update of the voter list, finalized on August 5, 2024, 17,140,354 voters are registered in Sri Lanka.

As we approach the 2024 general election, the fight for reliable polling data is more competitive than ever. The political landscape is rapidly shifting, Voter behavior has become more dynamic, making accurate predictions challenging for analysts relying on limited data sources. Phone and door-to-door surveys, once the bedrock of the election season, have seen significant declines in participation. Artificial intelligence has become a major tool that changes the way we gather and interpret massive amounts of data, providing more nuanced insights into what the electorate may be thinking.

Sinhala is a morphologically rich language that belongs to the Indo-Aryan branch of Indo-European languages. More than 16 million people(74% of the population) use the Sinhala language to communicate and more than 7 million people (32.1% of the population) use the Internet in Sri Lanka(Chathuranga et al., 2019). However, social media platforms have become a primary platform for political discourse, offering valuable insights into public opinion. However, the lack of tools to analyze political sentiment in the Sinhala language on these platforms poses a significant challenge.

Understanding public opinion is crucial for policymakers, enabling them to make informed decisions that align with the needs and concerns of the population. Political campaigns aim to connect with voters, but grasping the opinions of a large crowd can be difficult. Sentiment analysis functions as a listening tool, allowing politicians to understand what different segments of the population care about and how they respond to campaign messages. This insight enables them to tailor their communication strategies to resonate with specific groups, ensuring that their messages effectively engage the right audiences.

Consequently, this approach can lead to more effective and inclusive political campaigns. In addition, sentiment analysis can act as an early warning system for possible political dissatisfaction. By identifying new trends in public opinion, authorities can take preventive measures to address issues and maintain social stability. Therefore, recognizing the need to bridge the gap in Sinhala political sentiment analysis and its vital role in enhancing Sri Lanka's democracy, this research aims to develop a sentiment analysis model that incorporates Explainable Artificial Intelligence (XAI) techniques. Ultimately, this effort seeks to contribute to a more transparent and informed political landscape, empowering the Sri

Lankan public.

1.2 Problem Statement

There is a significant gap in studies that analyze public opinion, which is essential for policy makers to make informed decisions that address the needs and concerns of the people. Furthermore, existing approaches often face challenges due to limited resources for the Sinhala language.

1.3 Aims and Objective

The main objective of the research is to contribute to the field of Natural Language Processing (NLP) by developing a sentiment analysis model specifically for the Sinhala language in the domain of politics providing insights into political opinions expressed on social media. The specific objectives for the study are as follows :

- Contribute a data set for the community which can be used for further Sinhala text analysis regarding the recent presidential election 2019 and the parliamentary elections 2020.
- Develop an AI driven strategy to explain the election results by analyzing the newly scrapped youtube comments.
- Develop a sentiment analysis model to classify youtube comments.
- Leverage Explainable AI (XAI) techniques to interpret the political sentiment analysis model.

CHAPTER 2

LITERATURE REVIEW

Artificial intelligence is transforming virtually all aspects of social and business life. For better or worse, this includes political campaigns and elections.

An emotion is a complex feeling state in psychology that generates physical and psychological changes and influences human thought and behaviour. Words like "happy", "sad", "furious", "depressed", "love", "hate" and so on can be used to express emotions. The interpretation and comprehensive classification of emotions (positive, negative, and neutral) within text data using text analysis techniques are known as sentiment analysis. This research focused on classifying political sentiments using novel AI techniques.

2.1 Sentiment Analysis for Sinhala Language

Feature extraction plays a crucial role in achieving accurate classification of analysing sentiment in social media content. The study (Jayasuriya et al., 2020a) investigates the effectiveness of two common feature extraction techniques which are word n-grams and character n-grams. Their focus lies on sentiment analysis of Sinhala text, a language with unique morphological characteristics. The authors compared the performance of machine learning algorithms trained on features derived from both word and character n-grams. Furthermore, their findings suggest that character n-grams achieved superior performance, likely due to their ability to capture sentiment-specific morphemes in Sinhala, a language with rich morphology. This study highlights the importance of considering language-specific features when developing sentiment analysis models for social media data.

Furthermore, (Demotte et al., 2020) explored the application of deep learning for sentiment analysis in Sinhala, a low-resource and morphologically rich language. Prior research in this area primarily relied on traditional machine learning techniques. This study proposed a novel approach as Sentence-State Long Short Term Memory Networks (S-LSTMs) for

sentiment classification of Sinhala news comments. The authors mentioned that S-LSTMs are particularly well suited for capturing the sequential nature of language and the complex morphological features of Sinhala.

Researchers analysed sentiment in Sinhala social media text, particularly focused on the sports domain in (Jayasuriya et al., 2020b) study. Analysing sentiment in Sinhala poses a challenge due to the informal language used online. To address this, the study combined machine learning algorithms, which learn from labelled data, with lexicon-based methods that rely on pre-defined emotional associations of Sinhala words. Their evaluation of social media comments showed that this combined approach improved the accuracy of sentiment classification in Sinhala content. This research highlights the effectiveness of combining techniques for sentiment analysis in Sinhala, considering the unique features of the language and informal social media contexts.

(Chathuranga et al., 2019) proposed a novel approach for sentiment analysis for the Sinhala language. Sentiment analysis aims to determine the emotional tone of text data, and this research focuses on creating a lexicon specifically for Sinhala, a resource that has been limited for this language. The authors mentioned that traditional lexicon development methods are often manual and time-consuming. They proposed a corpus-based approach that leverages a large collection of Sinhala text to automatically identify words and assign sentiment polarities (positive, negative, or neutral). This method utilises techniques like analysing conjunctions within adjective pairs and morphological relationships to determine word polarity. The resulting sentiment lexicon is then employed for sentiment classification tasks. The study demonstrates that this corpus-based method has the potential to be more accurate with larger text corpora. This research contributes to the development of natural language processing (NLP) tools for Sinhala by providing a valuable resource for sentiment analysis tasks.

(Dhananjaya et al., 2022a) presents a comprehensive exploration of pre-trained language models (PLMs) for Sinhala text classification. The study addresses the scarcity of resources for Sinhala Natural Language Processing (NLP) by evaluating both multilingual and monolingual PLMs. The research encompasses a comparative analysis of several PLMs, in-

cluding Cross-Lingual Language Model with RoBERTa architecture(XLM-R), Language-Agnostic BERT Sentence Embedding (LaBSE), Language-Agnostic Sentence Representations (LASER), and two newly proposed monolingual models based on the RoBERTa architecture. These models are evaluated on a diverse set of Sinhala text classification tasks. Experimental results demonstrate the superior performance of the proposed monolingual models compared to existing multilingual options. Beyond model evaluation, the authors contribute to the Sinhala NLP community by releasing newly annotated datasets and pretrained models. Additionally, they provide practical recommendations for utilizing PLMs effectively in Sinhala text classification scenarios, considering factors such as data availability and computational resources.

2.2 Unsupervised Sentiment Classification

(Kapoor & Jindal, 2020) explored the application of Self-Organizing Maps (SOMs) for unsupervised sentiment classification in the domain of brand sentiment analysis on Twitter. While supervised learning approaches are prevalent in this area, this study highlights the advantages of SOMs. Compared to supervised models, SOMs offer the benefit of data visualisation. By mapping tweets onto a visual representation, SOMs allow researchers to identify clusters with distinct sentiment polarities towards a brand. This visual analysis can provide valuable insights alongside the sentiment classification. However, they mentioned that supervised learning might remain the preferred approach for tasks requiring the highest accuracy in sentiment classification.

Text data is a type of unstructured data, featuring high dimensions, large data volume, and low-value density. Traditional text clustering algorithms face two challenges, the high dimensionality of computing vectors and poor calculation efficiency. In another study, they developed a text clustering algorithm based on K-means clustering to address these challenges (Wang et al., 2019). Text data must be represented in a form that can be processed by a computer. The researchers used the Word2vec algorithm as the word embedding method for this research.

2.3 Explainable AI in Sentiment Analysis

Recent developments in machine learning have introduced models that approach human performance at the cost of increased architectural complexity. Efforts to make the rationales behind the models' predictions transparent have inspired an abundance of new explainability techniques. Provided with an already trained model, they compute saliency scores for the words of an input instance.(Atanasova, 2024)

The explainability of a machine learning model is usually inverse to its prediction accuracy, the higher the prediction accuracy, the lower the model explainability. In recent years, AI researchers have aimed to open the black box of neural networks and turn it into a transparent system. Explainable AI could help developers to improve AI algorithms, by detecting data bias, discovering mistakes in the models, and remedying the weakness. (Xu et al., 2019)

Artificial intelligence is implemented as a "black box" that just gives the output after a certain input but how it is achieved is not revealed. However, some machine learning models may not be intuitive or transparent, which may be complex for people to understand. In such cases, these models may lose their effectiveness.

2.4 Data Collection

Even though the initial plan of the research was to scrape tweets since the Twitter API is not freely accessible anymore we scrapped comments from YouTube. The suitable time frame was identified as the pre-election campaign period for the presidential election. Thereafter, a method was developed to scrape comments from YouTube. The 2024 Sri Lankan presidential election was the ninth presidential election and was held on 21 September 2024. Hence, YouTube comments were collected up to September 21, 2024. However, since YouTube channels are biased, the data set has a bias. To reduce this bias different types of YouTube playlists were selected. During the first phase of this study, around 30,000 comments were scraped.

2.4.1 YouTube Data API Client

Sinhala comments which are related to the Sri Lankan presidential election and parliamentary election were scrapped from YouTube API. Figure 1 shows a part of the dataset scrapped from YouTube. For this approach, famous political talks, breaking news, YouTube shorts and other politics related video playlists were selected from verified news channels in Sri Lanka. After creating a project in the Google Cloud Console YouTube API was enabled. Thereafter, the API key was enabled for authentication. The "googleapiclient.discovery" Python library was used to interact with the YouTube API client. The YouTube Data API uses a quota system to ensure that developers use the service as intended and do not create API clients that unfairly reduce service quality or limit access for others. Projects that enable the YouTube Data API have a default quota allocation of 10,000 units per day. Therefore, it took some time to collect a sufficient amount of comments. To extract a sufficient amount of data around 15 youtube APIs were generated using different email addresses.

2.5 Description of the Dataset

Video metadata such as title, description, uploaded date, channel name, video ID, author's email, and published time was extracted. Finally, the entire dataset was extracted to the local

machine in a structured format as a CSV file. The original dataset contains 7 columns including comment ID, video ID, text, time-stamp, author, author email and 'reply to' (reference for comment) column.

Attribute	Data Type	Description
Comment ID	String	Unique identifier for the comment.
Video ID	String	Identifier for the video.
Video Title	String	Title of the video associated with the comment.
Text	String	The content of the comment.
Timestamp	String	The date and time when the comment was posted.
Author	String	Username of the person who posted the comment.
Reply To	String	If the comment is a reply, the Comment ID of the original comment.

Table 2.1: Attributes of the dataset.

2.6 Preliminary Analysis

Figure 2.1 and 2.2 illustrates the channel statistics for the news channels that were used to scrape comments which is an indicator to get a basic idea of the popularity of the channels among people.

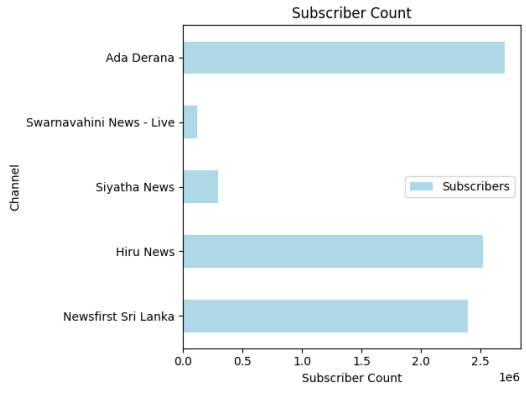

Figure 2.1: Subscriber Count

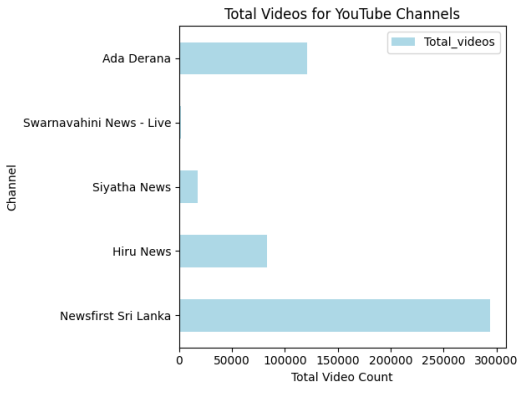

Figure 2.2: Video Count Per Channel

Furthermore, election results can be accessed from the https://results.elections.gov.lk official website. Therefore, the aim is to combine all available data sources to build an explainable AI approach to predict election results.

comment_id	video_id	video_title	text	timestamp	author
Ugw7OPfSzzRgrDkz4VV4AaABAg	PCTYPWpdhu0	අගමැතිගෙන් විශේෂ ප්‍රකාශයක්...	Very sad, why people doing this kind of things, 😢	2022-07-11T10:22:46Z	@ranjanihernetkoski2932
Ugw4MjfV0AFwOeBlmRN4AaABAg	PCTYPWpdhu0	අගමැතිගෙන් විශේෂ ප්‍රකාශයක්...	රනිල් කියන්නෙ හොද [illegible] [illegible] තමන්ගෙ ගෙව ගිනි...	2022-07-11T10:22:20Z	@dhanushkaprasad4342
UgyXQpMldLBQOoOMFpd4AaABAg	PCTYPWpdhu0	අගමැතිගෙන් විශේෂ ප්‍රකාශයක්...	හරි හරි ඔය [illegible] කතා වැඩක් නෑ මන්නෙ [illegible] ඒ...	2022-07-11T10:22:02Z	@kanishkagamage6628
Ugzs04SCAYmHEAvhinl4AaABAg	PCTYPWpdhu0	අගමැතිගෙන් විශේෂ ප්‍රකාශයක්...	යාපනය පුස්තකාලය	2022-07-11T10:21:49Z	@appachchidoniyai5731
Ugy80mi8AfnkCnh6Dlh4AaABAg	PCTYPWpdhu0	අගමැතිගෙන් විශේෂ ප්‍රකාශයක්...	මුලින් අයින් වෙලා යන්ඩ, ita පස්සේ pitatha ඉන්න...	2022-07-11T10:21:20Z	@roseperera2669
UgznYs37zBYnh1dnoOV4AaABAg	PCTYPWpdhu0	අගමැතිගෙන් විශේෂ ප්‍රකාශයක්...	මෙ කරන්නෙ මොකක්ද ඔයා....	2022-07-11T10:20:32Z	@abelashabhanu1767
UgxLIqiZh1ccBAuiDRJ4AaABAg	PCTYPWpdhu0	අගමැතිගෙන් විශේෂ ප්‍රකාශයක්...	කවුද [illegible] ආර්ථිකය කඩා වැටවුවේ අපි ද [illegible] ද 😂 [illegible]	2022-07-11T10:20:25Z	@kanishkagamage6628
UgznsxzTFZDKmG4H9sl4AaABAg	PCTYPWpdhu0	අගමැතිගෙන් විශේෂ ප්‍රකාශයක්...	අනේ යන්න මෙයා යන්න....ඔයා ගැහැන් අපි දන්නවා....	2022-07-11T10:20:17Z	@chrishanthieperera2500
UgyNEk_GkjVcqoliiwZ4AaABAg	PCTYPWpdhu0	අගමැතිගෙන් විශේෂ ප්‍රකාශයක්...	Hari boru meya kiynne	2022-07-11T10:20:05Z	@memeking7415

Figure 2.3: Sample Dataset

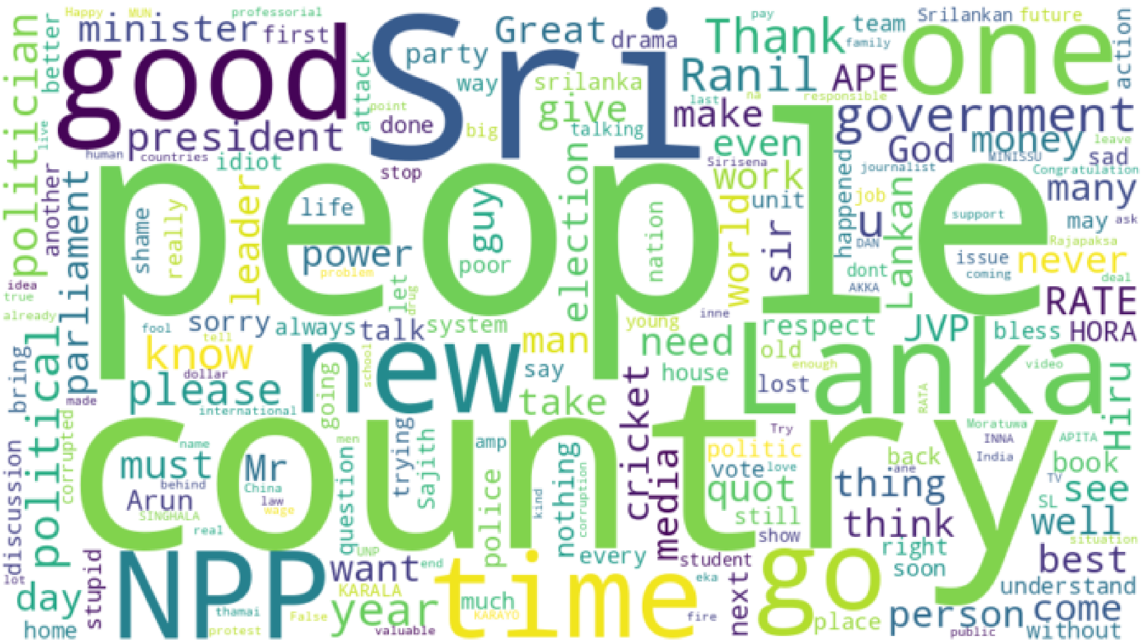

Figure 2.4: Wordcloud for English Comments

CHAPTER 3

METHODOLOGY

In this chapter, the strategy used for each process will be explained thoroughly. This chapter provides information to the readers about the sentiment analysis techniques, how the model was trained etc.

3.1 Dataset Preprocessing

After the data collection process was completed, the data set was manually reviewed to ensure the relevance and quality of the comments. During the manual review, the data set was checked for proper language and a balanced representation of political views.

The objective of text cleaning is to remove noise and irrelevant content from comments to improve the quality of the data. Text cleaning steps include removing HTML tags, special characters, and extra whitespaces. Further, eliminate URLs, email addresses, and numbers that are not meaningful in context. In this project, we used a language detection Python library, "langdetect", to filter out non-Sinhala comments. Thereafter, a sample was manually validated to confirm the accuracy.

The texts used in the comments section do not always employ standard language. Comments often contain slang and improper forms of words, making it difficult to extract features from them. Not all the comments posted were about the video or related to the channel. A large number of viewers comment to market their products or just to show their presence. These comments are not useful to the content creators and only add unnecessary overhead. These issues are common in platforms like YouTube because of the informal nature of communication.

3.2 Converting Texts into Features

3.3 Dataset Labeling - Manual Labeling

This manual labelling approach organizes data into a multi-stage classification pipeline to predict election results effectively. The dataset, consisting of a total of 10,000 comments, was evenly distributed among 10 individuals, with each person receiving 1,250 comments for labelling. To improve the reliability and consistency of the labelling process, a validation subset was created. Out of the 1,250 comments assigned to each individual, a sample of 250 comments was randomly selected and redistributed among three different participants. These three participants independently labelled the same sample, and the final label was determined using a majority voting system. This approach ensured that the final labels were based on consensus rather than individual interpretation, thereby reducing subjective errors.

Participants were explicitly advised to set aside their personal political opinions and focus solely on the objective content of the comments during the labelling process. This directive was aimed at minimizing cognitive and emotional biases that might influence the labelling outcome, especially given the political nature of the dataset.

Several strategies were employed to further reduce potential biases during the manual labelling process. Detailed and standardized instructions were provided to all participants to ensure a uniform understanding of the labelling criteria. This included specific examples and edge cases to clarify ambiguous scenarios. The labelling team was composed of individuals from diverse backgrounds to balance any inherent biases that might arise from homogeneous perspectives. Feedback was provided to participants throughout the process to correct any deviations from the labelling guidelines and ensure consistent application of criteria.

- Label 1: Political/Non-Political Content

  The first model (Model 1) identifies whether the content is political or non-political. This step filters out irrelevant data, focusing only on political content relevant to predicting election outcomes.

- Label 2: Positive/Negative/Neutral Sentiment Analysis

For data classified as political, the second model (Model 2) assesses the sentiment as positive, negative, or neutral. This helps gauge public perception or sentiment toward political entities, which is a critical factor in election predictions.

- Label 3: Entity Categorization

  The final model (Model 3) classifies political content into categories mentioning individual politicians or political parties. This provides granularity in understanding how sentiment is distributed among different political entities. As explained above, candidates included Ranil Wickramasinghe as the independent candidate, Leader of the Opposition Sajith Premadasa, Anura Kumara Dissanayake of the NPP, and Namal Rajapaksa, son of former President Mahinda Rajapaksa. Therefore as label 3, we had five labels which are Anura Kumara Dissanayake(NPP), Ranil Wickramasinghe, Sajith Premadasa, Namal Rajapaksha, Other category and Everyone category (All the politicians).

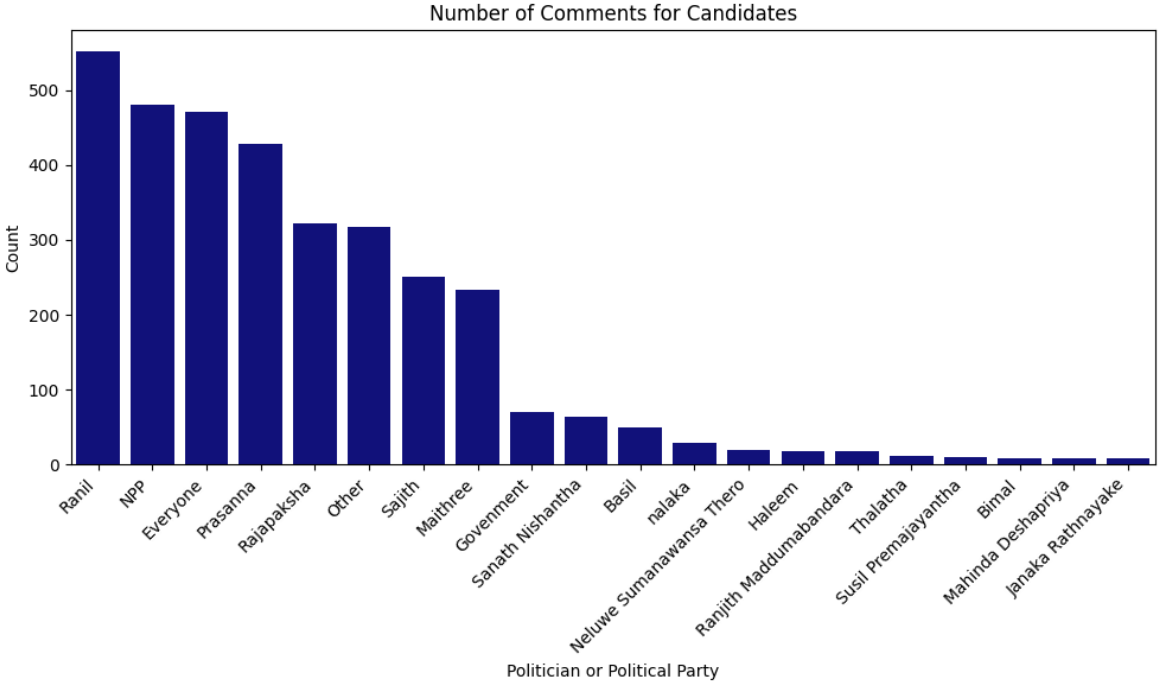

Figure 3.1: Total No.of Comments Per Candidate

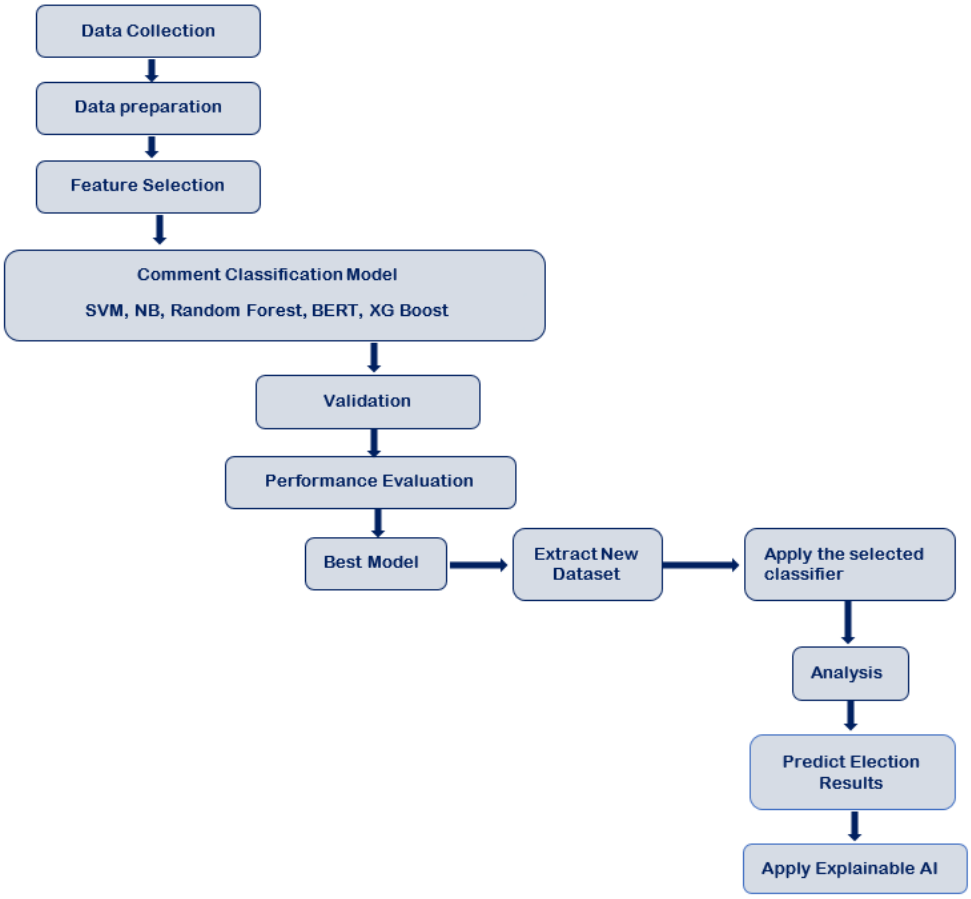

Figure 3.2: Methodology

3.4 Building Models for Multi-Stage Labeling

To address the classification of the dataset into three distinct labels, we developed three independent AI models. Each model was designed to specialize in identifying a specific label as explained below.

1. Model 1 :

   Classifying Political vs. Non-Political Content

   This binary classification model served as the foundational step. This model helps to filter out irrelevant data and ensure that analysis focuses solely on politically relevant comments. This model used textual features such as word embeddings, n-grams and linguistic patterns derived from the comments.

2. Model 2:

   Sentiment Analysis of Political Content

   Once a comment was classified as political, the second model identified the sentiment polarity as positive, negative or neutral. This multiclass classification step aimed to capture public sentiment toward political entities.

3. Model 3:

   Identifying Political Entities (Politician vs. Political Party)

   The third model categorized the political comments further into two subcategories references to each candidate or political party.

Each model was optimized for its specific task, improving overall performance and reducing errors that might arise. By combining traditional machine learning methods (SVM, Naive Bayes, Random Forest, XGBoost) with advanced deep learning techniques (BERT), this research achieved a balance between efficiency, accuracy, and scalability for text classification.

3.5 Feature Extraction

After the data are cleaned and preprocessed, data should be converted into a form that the model can understand. For this, all variables must be converted into numerical form. This process is called feature extraction or vectorization. This process also contributes to dimensionality reduction and, hence, helps with feature extraction, to keep only the features that improve the accuracy of the model. Feature extraction can be performed using methods. The importance of the words occurring in the dataset can be gauged, and redundant data can be removed. New features can also be formed from existing ones. Through such methods, features that matter and new features can be generated to form a better version of the original dataset. We used Count Vectorizer in this research, which is used for converting text into a vector. The TF-IDF (term frequency-inverse document frequency) statistic examines the relevance of a word to a document in a collection of documents.

3.6 Bidirectional Encoder Representation of Transformers (BERT)

The BERT model is a relatively new language model that was presented by Google in 2018. This model has presented state-of-the-art results in natural language processing. The key feature of BERT is the bidirectionality of the model. The BERT model makes use of the encoder component of the transformer to furnish the representation of words. BERT is used for the creation of language representation models that can serve various purposes. BERT has a base layer of "knowledge" that is derived from its pretraining. From this base layer of "knowledge", BERT can further be trained to adapt to the specifications provided. BERT's transformer processes any given word concerning the word's relation to all other words in that particular sentence. This enables BERT to understand the context of the word after looking at all surrounding words, unlike other models that understand the meaning of a word in one dimension only (Mehta & Passi, 2022).

The attention mechanism in BERT, specifically its multi-head self-attention, plays a crucial role in understanding the relationships between words in a sentence, regardless of their positions. This mechanism allows the model to focus on different parts of the input text simultaneously, capturing both local and global dependencies.(Vaswani, 2017)

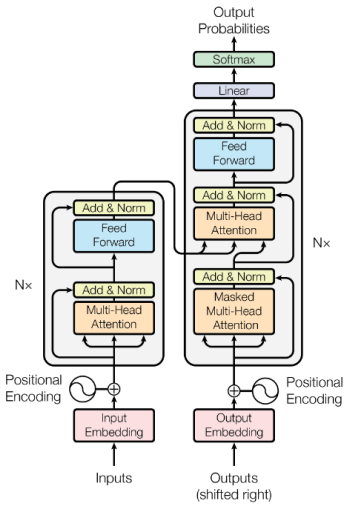

Figure 3.3: The Transformer Model Architecture

In this research, we utilized SinBERT, a pre-trained language model tailored for the Sinhala language, to classify YouTube comments for predicting election results. SinBERT is

based on the Bidirectional Encoder Representations from Transformers (BERT) architecture, which has significantly advanced natural language processing tasks by capturing deep contextual relationships within the text. (Dhananjaya et al., 2022b)


```
RobertaConfig {
  "_name_or_path": "NLPC-UOM/SinBERT-small",
  "architectures": [
    "RobertaForMaskedLM"
  ],
  "attention_probs_dropout_prob": 0.1,
  "bos_token_id": 0,
  "classifier_dropout": null,
  "eos_token_id": 2,
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 768,
  "initializer_range": 0.02,
  "intermediate_size": 3072,
  "layer_norm_eps": 1e-12,
  "max_position_embeddings": 514,
  "model_type": "roberta",
  "num_attention_heads": 6,
  "num_hidden_layers": 6,
  "pad_token_id": 1,
  "position_embedding_type": "absolute",
  "torch_dtype": "float32",
  "transformers_version": "4.42.4",
  "type_vocab_size": 1,
  "use_cache": true,
  "vocab_size": 30000
}
```


Figure 3.4: sinBERT Model Configuration

Figure 3.4 illustrates the model configuration for the sinBERT model used for this research.

3.7 Support Vector Machine

Support Vector Machine (SVM) is a powerful supervised machine learning algorithm widely used for classification and regression tasks, particularly effective in high-dimensional spaces such as text classification.

SVM is a classification algorithm that aims to find the optimal hyperplane that best separates data points of different classes. For a binary classification problem, the hyperplane is a decision boundary that divides the feature space into two regions, one for each class.

The algorithm works by maximizing the margin between the data points and the hyperplane, ensuring robustness and better generalization.

Text data, represented as vectors (e.g., TF-IDF, Bag-of-Words, or embeddings), typically has thousands of features. SVM performs well in high-dimensional spaces.

3.8 Random Forest

The Random Forest (RF) classifiers are suitable for dealing with the high dimensional noisy data in text classification(Islam et al., 2019). Random forest is an extension of ensemble learning algorithms (Bagging) that combines the output of multiple decision trees to reach a single result. Node size, the number of trees, and the number of features sampled are three primary hyperparameters, which need to be set before training the random forest algorithm. It can be used to solve regression or classification problems.

3.9 Naive Bayes

The Naive Bayes algorithm is a simple probability classifier that calculates a set of probabilities by counting the frequency and combinations of values in a given data set. The algorithm uses Bayes's theorem and assumes that all variables are independent considering the value of the class variable. This conditional independence assumption is rarely valid in real-world applications, so it is characterized as Naive, but the algorithm tends to learn quickly in a variety of controlled classification problems. Bayes' theorem is a mathematical formula used to determine conditional probability, which is named after 18th-century British mathematician Thomas Bayes. (Rish et al., 2001)

The Bayes theory is explained by the equation,

$$P(c|x) = \frac{P(x|c) \cdot P(c)}{P(x)} \tag{3.1}$$

$$P(c \mid x) = P(x_1 \mid c) \cdot P(x_2 \mid c) \cdot \ldots \cdot P(x_n \mid c) \cdot P(c) \tag{3.2}$$

Above,

- c, x are events.
- $P(c|x)$ is the posterior probability of class (c, target) given predictor (x, attributes).
- $P(x|c)$ is the likelihood, which is the probability of the predictor given class.
- $P(c)$ is the prior probability of class.
- $P(x)$ is the prior probability of the predictor.

3.10 XG Boost Classifier

Similar to Random Forest, XGBoost is an ensemble classifier made of decision trees and a variant of the Tree gradient boosting algorithm (El Rifai et al., 2022). The boosting method focuses on the new learning process on data with a weak learner value than the previous process(Syahrani, 2019).

3.11 SMOTE

The challenge of working with imbalanced datasets is that most machine learning techniques will ignore, and in turn have poor performance on, the minority class, although typically it is the performance of the minority class that is most important. One approach to addressing imbalanced datasets is to oversample the minority class. The simplest approach involves duplicating examples in the minority class, although these examples don't add any new information to the model. Instead, new examples can be synthesized from the existing examples. This is a type of data augmentation for the minority class and is referred to as the SMOTE. Simply SMOTE is an over-sampling approach in which the minority class is oversampled by creating "synthetic" examples rather than by over-sampling with replacement.(Chawla et al., 2002)

3.12 Explainable AI

There are several methods for generating explanations for model predictions, including:

1. LIME (Local Interpretable Model-agnostic Explanations) introduced by (Ribeiro et al., 2016) which generates explanations by perturbing the input and measuring the change in the model's output.

2. SHAP (SHapley Additive exPlanations) which uses a cooperative game theory to explain the output of any model.

3. Anchors automatically identify and characterize the features of input instances that are most indicative of a particular prediction.

4. Gradient-based methods that use the gradients of the output concerning the input to identify the most important input features for a given prediction.

In this research, we explored LIME (Ribeiro et al., 2016) for Sinhala text classification along with various machine learning algorithms such as support vector machine(SVM), XGBoost, Naive Bayes and Random Forest.

3.13 Explainable AI for BERT

BertViz is an interactive tool for visualizing attention in Transformer language models such as BERT, GPT2, or T5. It can be run inside a Jupyter or Colab notebook through a simple Python API that supports most Huggingface models. BertViz extends the Tensor2Tensor visualization tool by Llion Jones, providing multiple views that each offers a unique lens into the attention mechanism.

3.14 Evaluation Metrics

In the evaluation of the performance of classifiers, Confusion Matrix, Accuracy, F1-score, Recall and Precision are used.

3.14.1 Confusion Matrix

The Confusion Matrix is a tabular representation of Actual vs Predicted values. It has 4 quadrants and each of the observation data points in a classification problem belongs to one

of the possible four outcomes as shown in Figure 3.5. In the context of this project, TP represents the number of political comments correctly identified as political, TN represents the number of non-political cases identified as non-political, FP represents the number of non-political comments incorrectly identified as political, and FN represents the number of political cases incorrectly identified as non-political.

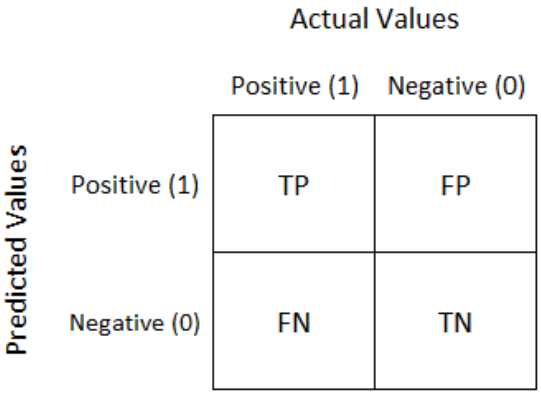

Figure 3.5: The Confusion Matrix

3.14.2 Accuracy

Accuracy is the most popular performance measure in the data community. It is simply a the ratio of correctly predicted observations to the total observations.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{3.3}$$

3.14.3 Precision

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

$$\text{Precision} = \frac{TP}{TP+FP} \tag{3.4}$$

3.14.4 Recall/Sensitivity

Recall can be described as the ratio between the number of positive predictions and the number of positive class values.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3.5}$$

3.14.5 F1-Score

The F1 Score is the weighted average of Precision and Recall. It tries to find the balance between precision and recall. This is very useful and not misleading compared to accuracy especially when dealing with imbalance data sets.

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.6}$$

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Unsupervised Clustering Method

Before the manual labelling approach was completed, unsupervised learning techniques were tested as the initial method. In the context of text classification using K-Means clustering, texts are classified into categories based on their feature representations and their assignment to clusters. Text data is converted into numerical form using Word2Vec word embeddings. Each text is represented as a vector in a high-dimensional space using word embeddings. Principal Component Analysis (PCA) was used to reduce the dimensionality of the feature space which makes the data easier to visualize and helps eliminate noise. K-means assign each text vector to one of the k clusters by minimizing the distance between the data points and cluster centroids. The k value is determined by the elbow method.

According to the output obtained, the visualized clusters (based on PCA components) show distinct groupings, which means the algorithm identified separable groups of similar texts. To assign meaning to these clusters, we would need to analyze representative samples from each cluster. After analysing the features of each text we failed to identify a similarity between texts in the same cluster or difference within different clusters.

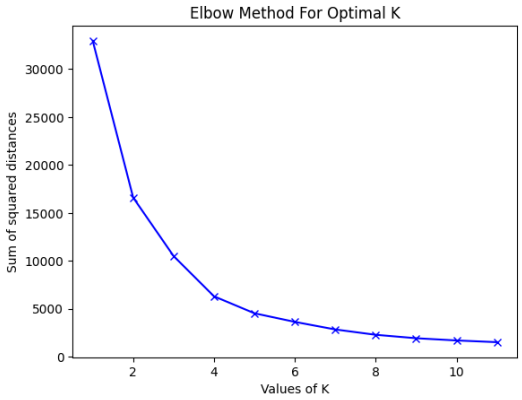

Figure 4.1: Elbow Curve

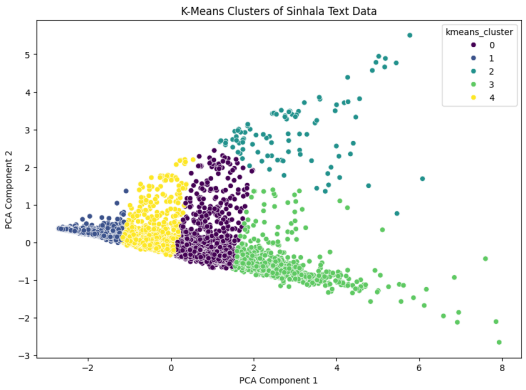

Figure 4.2: K-Means Clusters

Since unsupervised learning techniques lacked accuracy, we chose the more time-consuming

approach of manual labelling.

4.2 Supervised Classification Model

In this phase, the model evaluation will be discussed further for supervised classification. The results of the text classification models reveal significant insights into the effectiveness of various algorithms for predicting election results based on YouTube comments.

4.2.1 Model 1: Political/Non-Political Model

4.2.2 Naive Bayes Algorithm

The performance evaluation matrix for the Multinomial Naive Bayes Classifier is shown below. The AUC was recorded as 0.89, the accuracy of the model was recorded as 0.81, the precision of the model was 0.81 the recall for the model was found to be 0.81, and the f-measure also known as the F1 Score was recorded at 0.81.

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

           0       0.86      0.76      0.81       422
           1       0.76      0.86      0.81       374

    accuracy                           0.81       796
   macro avg       0.81      0.81      0.81       796
weighted avg       0.81      0.81      0.81       796

Confusion Matrix:
 [[319 103]
 [ 51 323]]
```

Figure 4.3: Classification Report

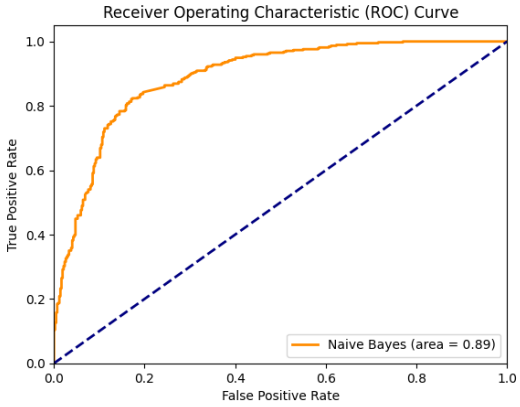

Figure 4.4: ROC curve

4.2.3 Support Vector Machine

The SVM model for classifying political and non-political comments shows strong performance. The ROC curve has an area under the curve (AUC) of 0.89, indicating excellent discriminatory ability. From the confusion matrix, the model correctly classifies 346 non-political comments and 304 political comments, with a total of 76 false positives (non-

political misclassified as political) and 70 false negatives (political misclassified as non-political). Overall, the model demonstrates a good balance between precision and recall.

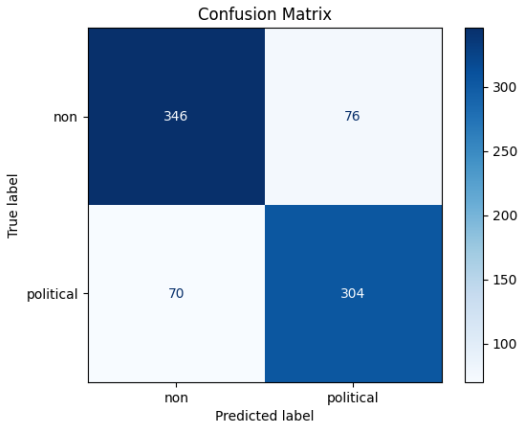

Figure 4.5: Confusion Matrix for SVM

	precision	recall	f1-score	support
non	0.83	0.82	0.83	422
political	0.80	0.81	0.81	374
accuracy			0.82	796
macro avg	0.82	0.82	0.82	796
weighted avg	0.82	0.82	0.82	796

Figure 4.6: Classification Report for SVM

4.2.4 Random Forest Classifier

The Random Forest model demonstrates strong performance, with an AUC of 0.90, reflecting its ability to effectively distinguish between political and non-political comments. The confusion matrix shows that 357 non-political comments and 306 political comments were correctly classified. However, the model misclassified 65 non-political comments as political (false positives) and 68 political comments as non-political (false negatives). Overall, the model performs well, achieving a good balance between capturing true positives and minimizing errors.

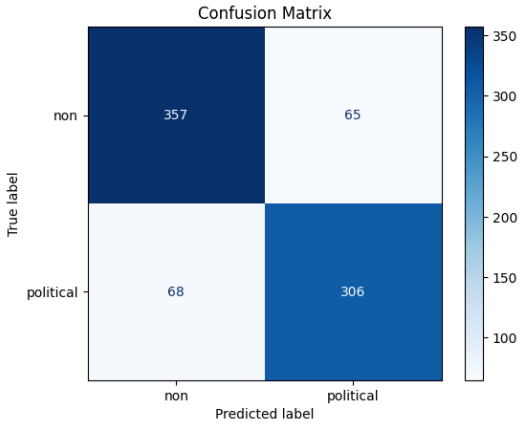

Figure 4.7: Confusion Matrix for Random Forest

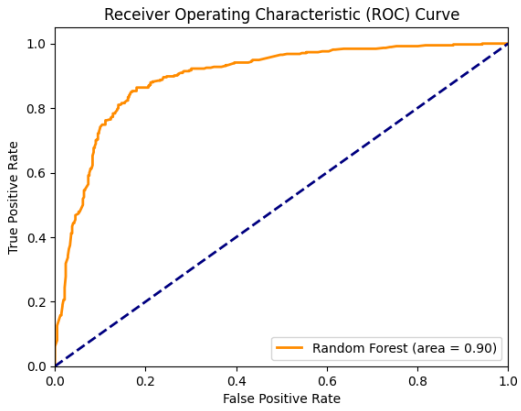

Figure 4.8: ROC curve for Random Forest

4.2.5 XG Boost Classifier

XG Boost classifier resulted in 360 true negatives and 306 true positives. 62 false positives and 68 false negatives. This translates to an accuracy of 0.83, a precision of 0.83, a recall of 0.83, and an f1 score of 0.83. The area under the ROC curve was 0.90. These results were obtained after using the random undersampling technique to handle the class imbalance problem.

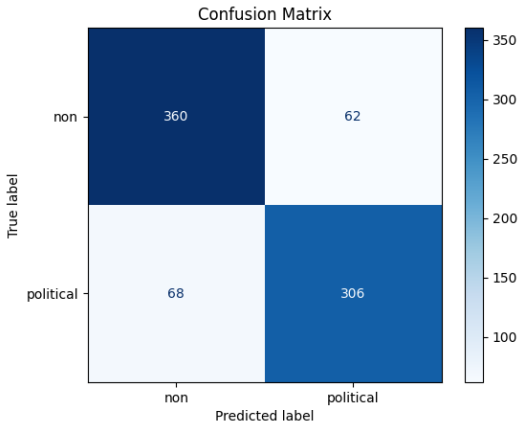

Figure 4.9: Confusion Matrix for XG-Boost

	precision	recall	f1-score	support
non	0.84	0.84	0.84	422
political	0.82	0.82	0.82	374
accuracy			0.83	796
macro avg	0.83	0.83	0.83	796
weighted avg	0.83	0.83	0.83	796

Figure 4.10: Classification Report for XGBoost

The LIME output for the XGBoost figure 4.11 indicates that the input text has been classified as political with 0.78 probability compared to 0.22 probability with Non-Political class. The most influential features contributing to this classification are shown by the orange bars in the feature importance graph. Conversely, words that contribute negatively to the "Political" prediction, supporting the "Non-Political" class are highlighted in blue. But, their impact is relatively smaller. The interpretation indicates that the model relies heavily on specific words to make its classification.

Figure 4.11: LIME Output for XGBoost

4.2.6 SinBERT model

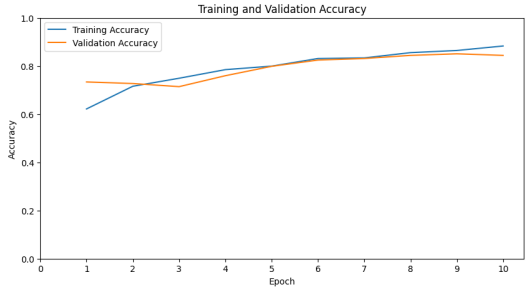

Figure 4.12: Accuracy Curve for SinBERT Model

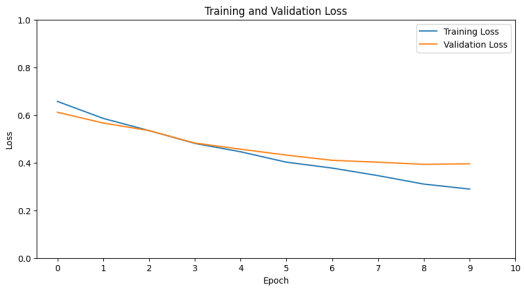

Figure 4.13: Loss Curve for SinBERT Model

	precision	recall	f1-score	support
Political	0.84	0.75	0.79	68
Non Political	0.82	0.88	0.85	86
accuracy			0.82	154
macro avg	0.83	0.82	0.82	154
weighted avg	0.83	0.82	0.82	154

Figure 4.14: Classification Report for SinBERT Model

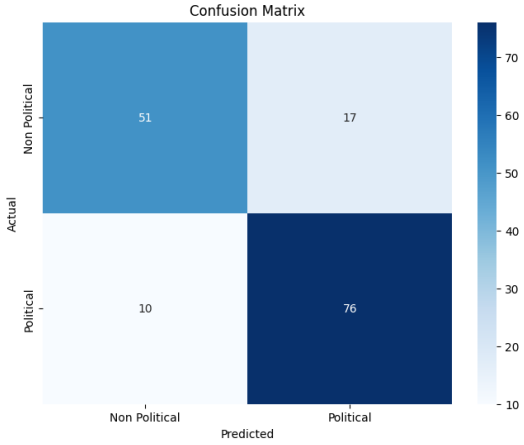

Figure 4.15: Confusion Matrix for SinBERT Model

The results from the SinBERT model demonstrate its effectiveness in classifying political and non-political comments. The accuracy curve in figure 4.12 shows steady improvement in both training and validation accuracy over 10 epochs, with no significant overfitting. Similarly, the loss curve in figure 4.13 reflects a consistent decrease in training and validation loss, indicating that the model is learning effectively. The classification report in figure 4.14 highlights strong performance, with an overall accuracy of 82% and macro-average precision, recall, and F1-score of 83%, 82%, and 82%, respectively. The confusion matrix (Figure 4.15) shows that the model correctly classified 127 out of 154 samples, but it misclassified 27 samples (17 non-political as political and 10 political as non-political). The precision and recall scores suggest that while the model performs well in identifying both classes, there is

a slightly better recall for non-political comments and slightly better precision for political comments. These results indicate that SinBERT is a robust model for this task.

4.2.7 AshenBERTo model

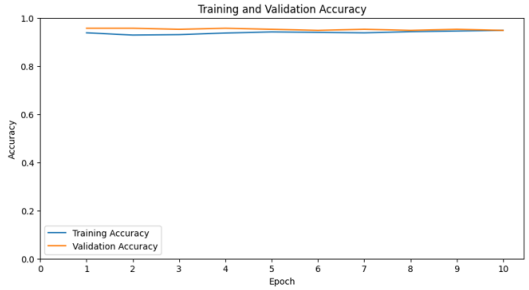

Figure 4.16: Accuracy Curve AshenBERTo Model

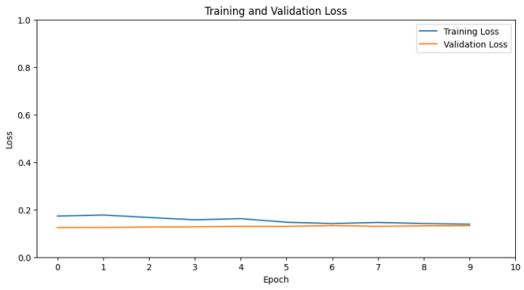

Figure 4.17: Loss Curve AshenBERTo Model

	precision	recall	f1-score	support
Political	0.95	0.92	0.94	119
Non Political	0.92	0.95	0.93	112
accuracy			0.94	231
macro avg	0.94	0.94	0.94	231
weighted avg	0.94	0.94	0.94	231

Figure 4.18: Classification Report

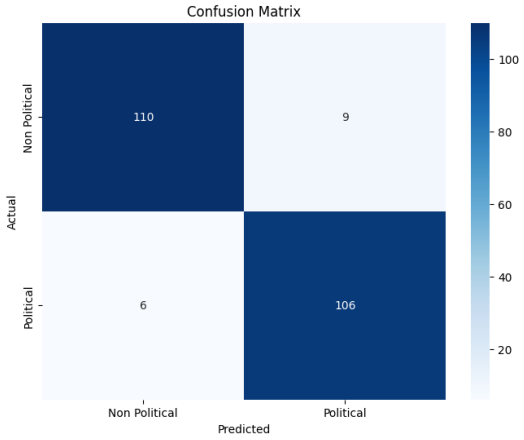

Figure 4.19: Confusion Matrix

The AshenBERTo model demonstrates superior performance compared to the sinBERT model across multiple evaluation metrics. The accuracy curve for AshenBERTo shows near-perfect alignment between training and validation accuracy, indicating minimal overfitting, while the sinBERT model exhibits a gap between the two, suggesting overfitting (Figures 4.12 vs. 4.16). Similarly, AshenBERTo's loss curve shows stable and smooth convergence, whereas sinBERT's validation loss displays slight instability (Figures 4.13 vs. 4.17). In terms of classification metrics, AshenBERTo achieves higher precision, recall, and F1-score for both political and non-political classes, with an overall accuracy of 94%, compared to sinBERT's

82% accuracy (Figures 4.18 vs. 4.14). Furthermore, AshenBERTo's confusion matrix reflects better classification performance with fewer misclassifications in both classes (Figure 4.19 vs. 4.15). Overall, AshenBERTo demonstrates stronger generalization and predictive capabilities, making it the preferred model.

4.2.8 Summary - Model 1

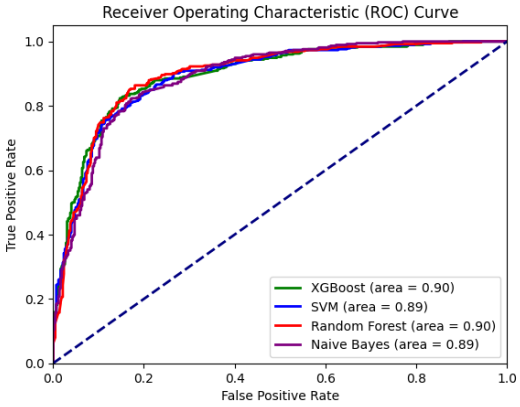

Figure 4.20: Comparison of ML models for Label 1

Figure 4.20 illustrates a comparison of ROC curves for model 1, which is built to identify irrelevant comments from political comments. XGBoost and Random Forest offer the best overall performance, as indicated by their AUC values and proximity to the top-left corner of the plot.

4.3 Model 2 : Negative/Positive

4.3.1 Random Forest model

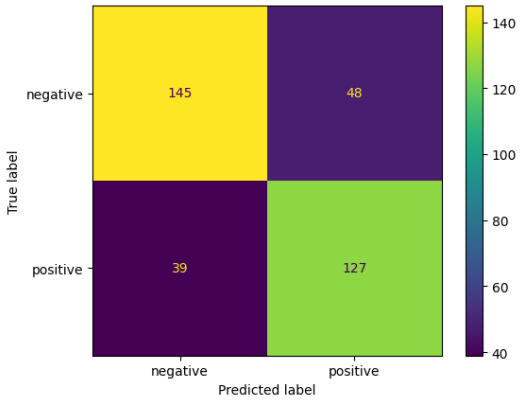

Figure 4.21: Confusion Matrix for Random Forest

Classification Report:

	precision	recall	f1-score	support
negative	0.79	0.75	0.77	193
positive	0.73	0.77	0.74	166
accuracy			0.76	359
macro avg	0.76	0.76	0.76	359
weighted avg	0.76	0.76	0.76	359

Figure 4.22: Classification Report for Random Forest

According to figure 4.30 area under the curve is 0.84, which indicates good performance. The confusion matrix shows the model's ability to correctly classify 145 negative and 127 positive instances with 48 false positives and 39 false negatives from the test dataset. These results indicate a strong balance between sensitivity and specificity.

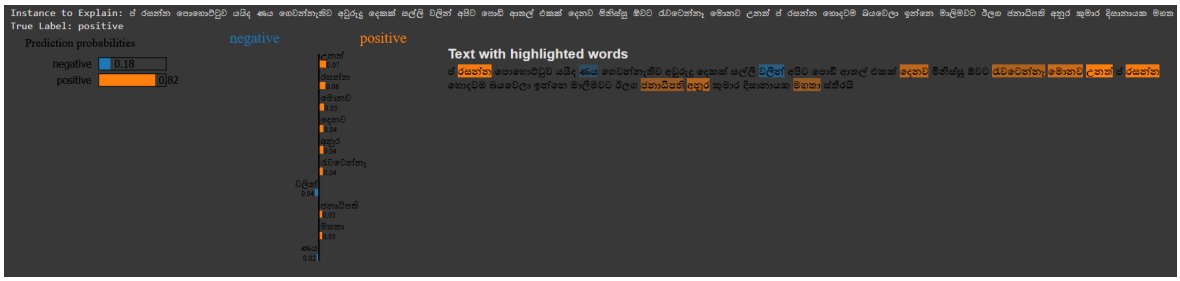

Figure 4.23: LIME Output for Random Forest

According to Figure 4.23 the positive prediction probability is 0.82 and the key contributing words of the input texts were highlighted to show their influence on the prediction.

4.3.2 Support Vector Machine

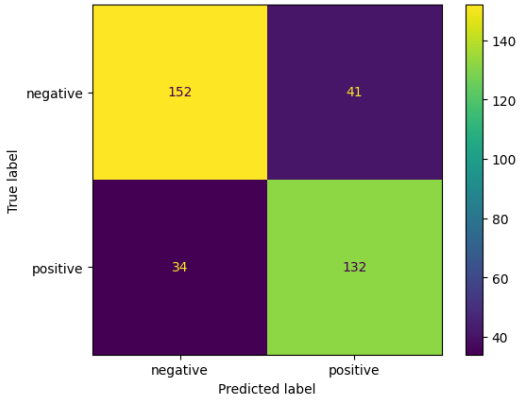

Figure 4.24: Confusion Matrix for SVM

SVM Classification Report:

	precision	recall	f1-score	support
negative	0.82	0.79	0.80	193
positive	0.76	0.80	0.78	166
accuracy			0.79	359
macro avg	0.79	0.79	0.79	359
weighted avg	0.79	0.79	0.79	359

Figure 4.25: Classification Report SVM

The Support Vector Machine (SVM) model achieved an overall accuracy of 79%, demonstrating its effectiveness in classifying the data into "negative" and "positive" classes. The confusion matrix indicates that the model correctly classified 152 "negative" and 132 "positive" instances while misclassifying 41 "negative" instances as "positive" and 34 "positive" instances as "negative." The classification report highlights a precision of 82% and a recall of 79% for the "negative" class, alongside a precision of 76% and a recall of 80% for the "positive" class, resulting in F1-scores of 0.80 and 0.78, respectively. Both the macro-average and weighted-average F1-scores are 0.79, reflecting balanced performance across classes. Despite its overall reliability, the model exhibits slightly lower precision for the "positive" class, suggesting potential areas for improvement in minimizing false positives and false negatives.

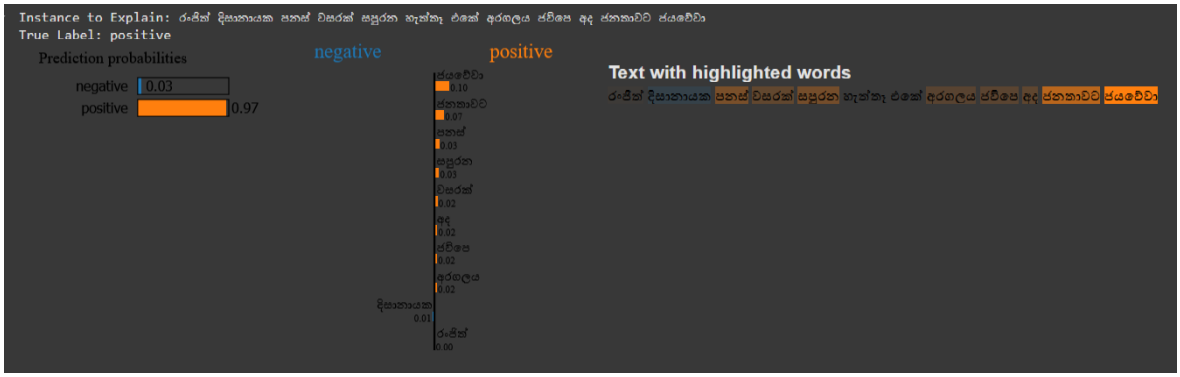

Figure 4.26: LIME Output for SVM

The LIME output provides insights into the decision-making process of the SVM model for a specific instance, which was correctly classified as "positive" with a probability of 0.97. The text highlights keywords and phrases that influenced the model's decision, with the colour intensity corresponding to their contribution. Words that are prominently highlighted, indicate their strong association with the "positive" class. On the other hand, no significant features contributed to the "negative" class, as its predicted probability was only 0.03. This visualization validates the model's reliance on meaningful linguistic patterns for classification, emphasizing the importance of these highlighted terms in predicting the sentiment accurately. The use of LIME enhances the interpretability of the model, providing transparency in understanding how individual predictions are made.

4.3.3 Naive Bayes Model

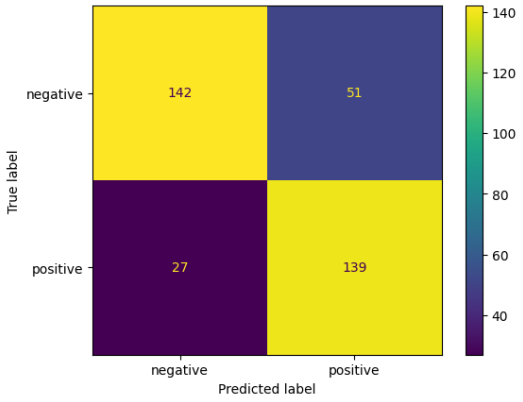

Figure 4.27: Confusion Matrix for NB

```
Naive Bayes Accuracy: 0.7827298050139275

Naive Bayes Classification Report:
              precision    recall  f1-score   support

    negative       0.84      0.74      0.78       193
    positive       0.73      0.84      0.78       166

    accuracy                           0.78       359
   macro avg       0.79      0.79      0.78       359
weighted avg       0.79      0.78      0.78       359
```

Figure 4.28: Classification Report NB

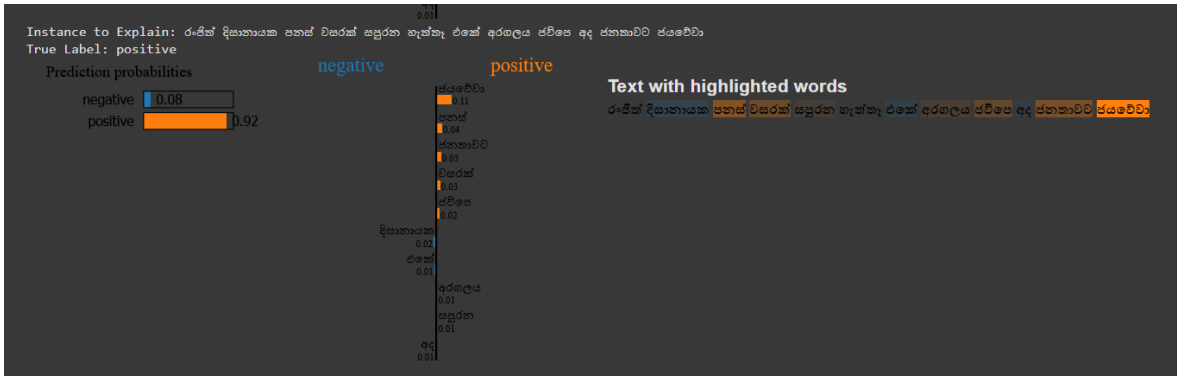

Figure 4.29: LIME Output for Naive Bayes Classifier

The Naive Bayes classifier achieved an overall accuracy of 78%, demonstrating its capability in classifying "negative" and "positive" instances. The model performed well for both classes, achieving an F1 score of 0.78 for each. However, it exhibited a slightly higher precision for the "negative" class (84%) and a higher recall for the "positive" class (84%), indicating a trade-off between false positives and false negatives. The LIME output (figure 4.29) provided interpretability by highlighting keywords, which contributed significantly to the model's correct prediction of a "positive" instance with 92% confidence.

4.3.4 Summary - Model 2

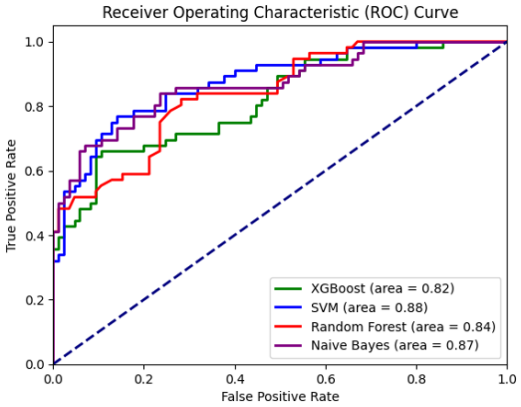

Figure 4.30: ROC curve for Model 2

4.3.5 sinBERT model

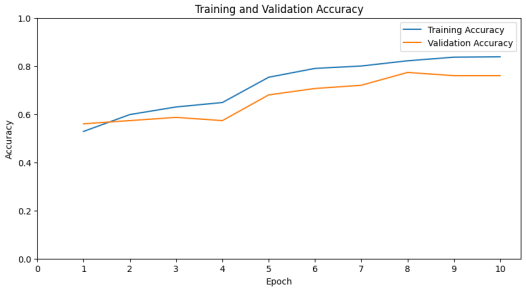

Figure 4.31: Accuracy Curve

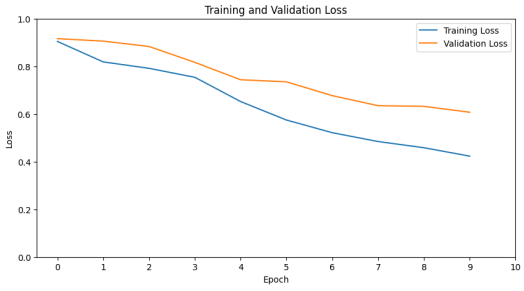

Figure 4.32: Loss Curve

4.3.6 AshenBERTo model

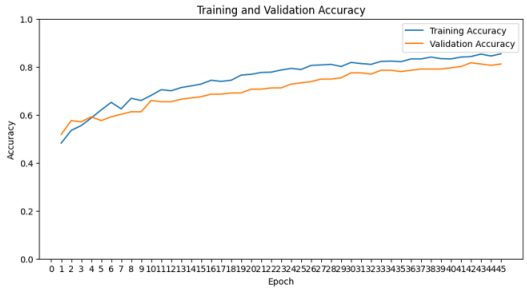

Figure 4.33: Accuracy Curve

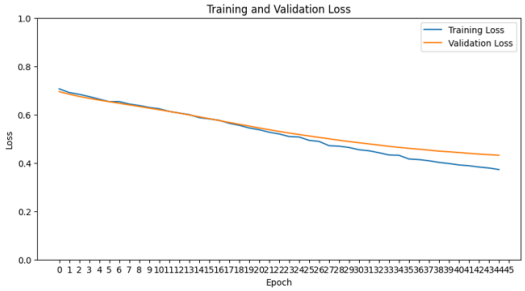

Figure 4.34: Loss Curve

Based on the above outputs, the sinBERT and AshenBeto models demonstrate good performance trends during training and validation, but a visible gap between training and validation accuracy suggests slight overfitting. Similarly, its loss curve (Figure 4.32) shows declining trends, yet with some fluctuation in validation loss, potentially indicating less stable generalization. In contrast, the AshenBERTo model (Figure 4.33 and 4.34) exhibits a more consistent alignment between training and validation accuracy, reflecting better generalization capabilities. Its loss curve also displays smoother and more stable convergence. Overall, while both models perform well, AshenBERTo demonstrates superior generalization and stability compared to sinBERT, making it a more robust choice for the given task.

4.4 Model 3 : Politician / Political Party

4.5 XGBoost Algorithm

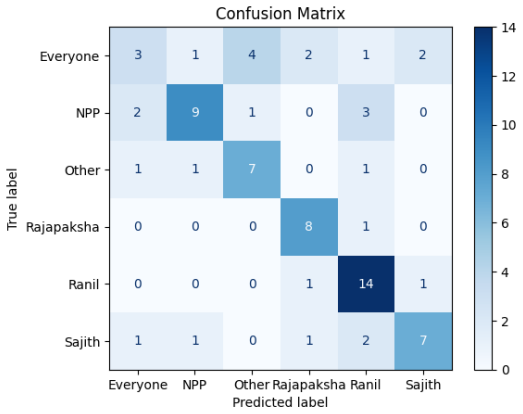

Figure 4.35: Confusion Matrix for XGBoost

	precision	recall	f1-score	support
Everyone	0.45	0.53	0.49	53
NPP	0.63	0.59	0.61	56
Other	0.53	0.58	0.55	45
Rajapaksha	0.70	0.56	0.63	55
Ranil	0.75	0.85	0.80	47
Sajith	0.72	0.66	0.69	47
accuracy			0.62	303
macro avg	0.63	0.63	0.63	303
weighted avg	0.63	0.62	0.62	303

Figure 4.36: Classification Report for XGBoost

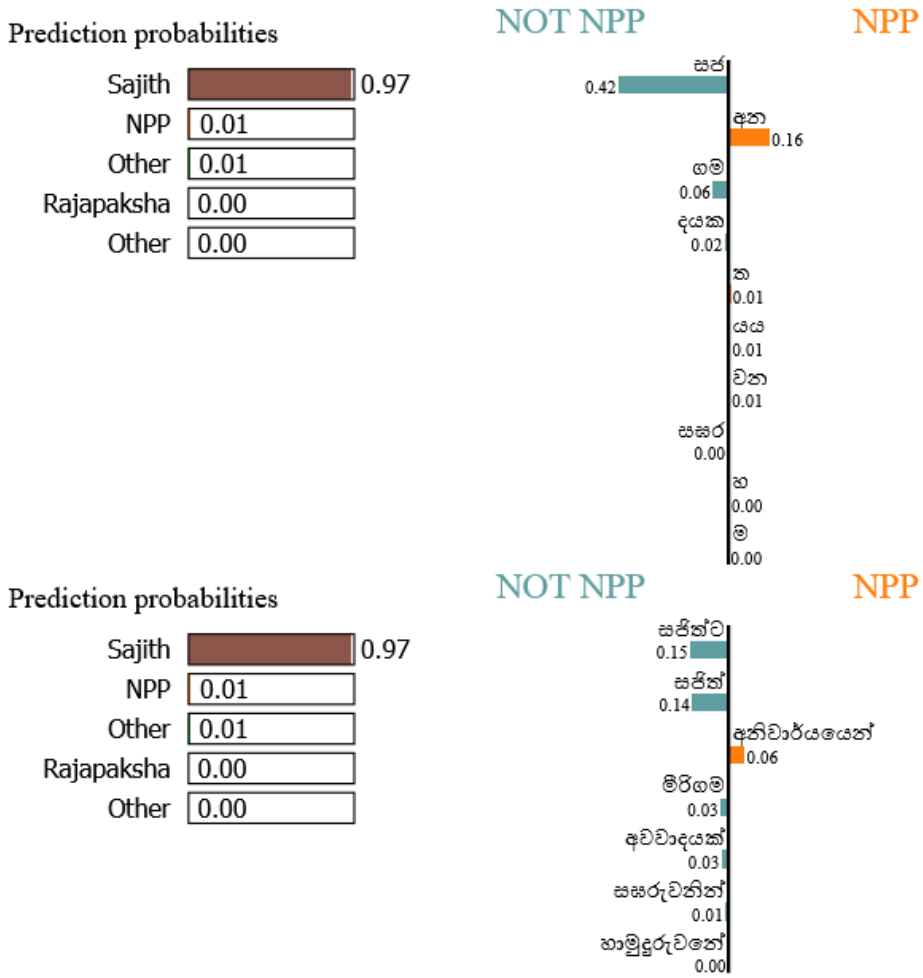

Figure 4.37: LIME Output for XGBoost

Figure 4.37 illustrates Explainable AI results obtained from two different tokenization techniques which are Sinlingua python library and Sinling python library. Sinling splits Sinhala words into separate words instead of meaningless tokens. However, both methods resulted in the same prediction probabilities for the "Sajith" category.

4.6 Support Vector Machine

In Figure 4.38, the confusion matrix provides information on the model's ability to correctly classify instances between different classes. The diagonal elements indicate the correct predictions, while the off-diagonal elements indicate misclassifications. The SVM model achieved a notable number of correct predictions for the "Ranil" and "Other" categories. However, some of the categories are misclassified. This may be due to overlapping features

or linguistic similarities in the dataset.

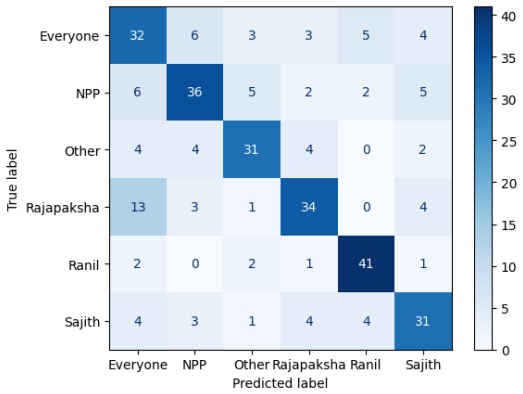

Figure 4.38: Confusion Matrix for SVM

	precision	recall	f1-score	support
Everyone	0.52	0.60	0.56	53
NPP	0.69	0.64	0.67	56
Other	0.72	0.69	0.70	45
Rajapaksha	0.71	0.62	0.66	55
Ranil	0.79	0.87	0.83	47
Sajith	0.66	0.66	0.66	47
accuracy			0.68	303
macro avg	0.68	0.68	0.68	303
weighted avg	0.68	0.68	0.68	303

Figure 4.39: Classification Report for SVM

The high precision for "Ranil", which is about 0.79, and "Other", which is about 0.72 indicates that the model is effective in identifying relevant instances for these classes while minimizing false positives. However, the lower precision for the "Everyone" category (0.52) suggests that it is a challenge for this model to differentiate this category from others. The overall weighted F1-score of 0.68 indicates general performance.

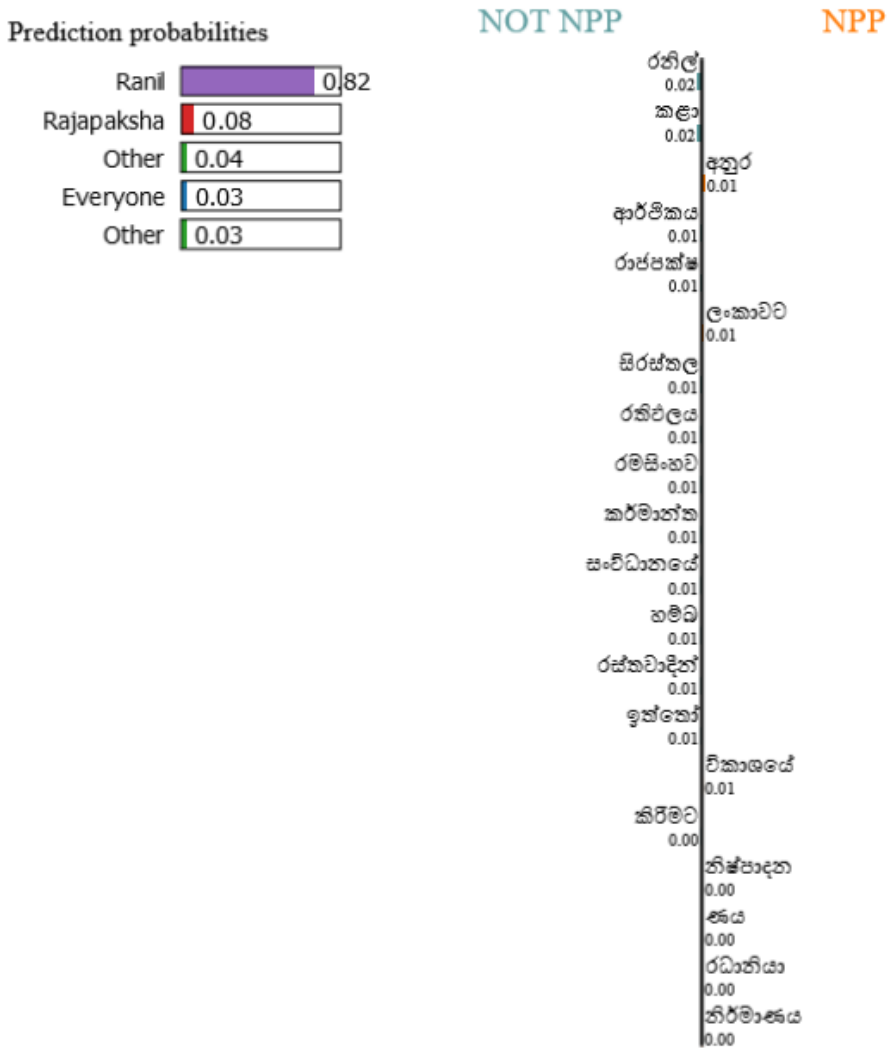

Figure 4.40: LIME Output for SVM

In figure 4.40 LIME explanation visualizes the contribution of individual words or tokens to the predicted class. In the Left side of the figure 4.40 shown as "NOT NPP", words contribute to reducing the likelihood of the text being classified under "NPP". Right-side "NPP" words increase the likelihood of the text being classified under "NPP". The model predicts the highest probability (82%) for the class "Ranil". This suggests the model is highly confident that the input text is associated with the class "Ranil".

4.6.1 Random Forest

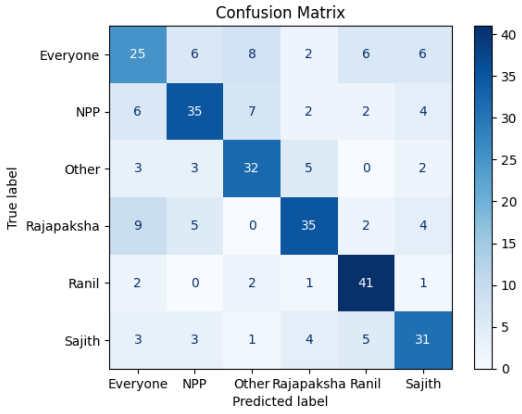

Figure 4.41: Confusion Matrix for Random Forest

	precision	recall	f1-score	support
Everyone	0.52	0.47	0.50	53
NPP	0.67	0.62	0.65	56
Other	0.64	0.71	0.67	45
Rajapaksha	0.71	0.64	0.67	55
Ranil	0.73	0.87	0.80	47
Sajith	0.65	0.66	0.65	47
accuracy			0.66	303
macro avg	0.65	0.66	0.66	303
weighted avg	0.65	0.66	0.65	303

Figure 4.42: Classification Report for Random Forest

In figure 4.42, the classification report provides a detailed summary of precision, recall, F1-score, and support for each class. For instance, the "Ranil" category exhibits the highest performance with a precision of 0.73, recall of 0.87, and an F1-score of 0.80, indicating that this class is well-identified by the model. In contrast, the "Everyone" class has lower precision (0.52) and recall (0.47), indicating difficulty in accurately predicting this category. The overall accuracy of the model is 66%, with macro and weighted averages for precision, recall, and F1-scores around 65%, reflecting moderate performance overall. In conclusion, while the model demonstrates satisfactory performance for certain classes, it faces challenges with others, particularly "Everyone" and "Rajapaksha."

4.6.2 Naive Bayes Model

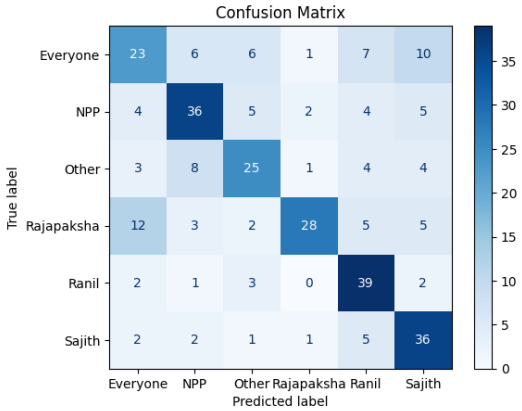

Figure 4.43: Confusion Matrix for Naive Bayes

	precision	recall	f1-score	support
Everyone	0.50	0.43	0.46	53
NPP	0.64	0.64	0.64	56
Other	0.60	0.56	0.57	45
Rajapaksha	0.85	0.51	0.64	55
Ranil	0.61	0.83	0.70	47
Sajith	0.58	0.77	0.66	47
accuracy			0.62	303
macro avg	0.63	0.62	0.61	303
weighted avg	0.63	0.62	0.61	303

Figure 4.44: Classification Report for Naive Bayes

Classification report for Naive Bayes in figure 4.44 illustrates that the "Ranil" class has a high recall of 0.83, meaning it captures most instances of this class, but the precision is slightly lower at 0.61, showing that it sometimes misclassifies instances as "Ranil." "Other" category has a precision of 0.60 and recall of 0.56, which indicates a moderate performance in distinguishing this class from others. The "Everyone" class has lower precision (0.50) and recall (0.43), suggesting that this category is challenging for the model. The overall accuracy of the Naive Bayes model is 62%, with macro and weighted average precision, recall, and F1-scores all around 0.63, 0.62 and 0.61. This reflects moderate performance, suggesting that while the model performs reasonably well on certain classes.

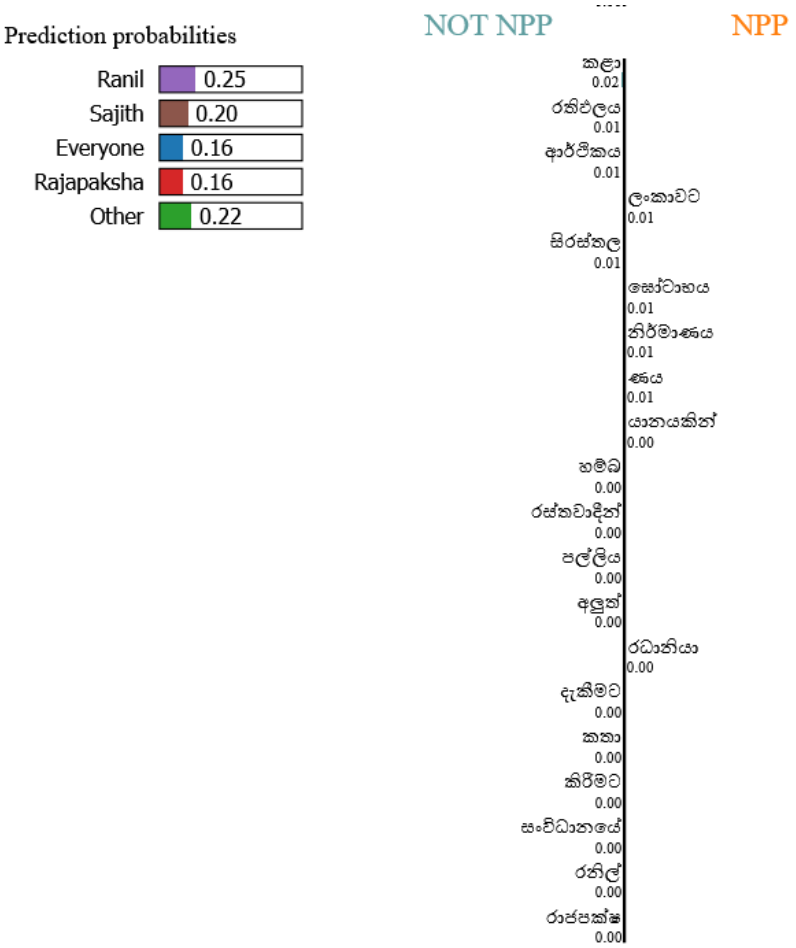

Figure 4.45: LIME Output for Naive Bayes

In figure 4.45 Words on the "NPP" side positively influence predictions for this class, aligning with the model's decent recall (0.64) for "NPP." However, the relatively balanced distribution of features near zero in the LIME visualization reflects the model's struggles with certain predictions, as evident from misclassifications in the confusion matrix (e.g., predicting "Sajith" as "Ranil"). This analysis highlights that while Naive Bayes relies on specific word features for predictions, it faces challenges due to overlapping feature importance across classes.

4.6.3 sinBERT Model

The results shown below are a comprehensive overview of the sinBERT model's performance. The confusion matrix indicates that the classification model performs well, with high precision across all categories. The diagonal dominance reflects strong predictions, particu-

larly for "Everyone" (50 correct predictions), "Rajapaksha" (49), and "Ranil" (45). There are no misclassifications, except for a few minor predictions seen for "NPP" (14) and "Sajith" (9). From the training loss and validation loss graph, the model shows steady convergence, with both losses gradually decreasing over 100 epochs and stabilizing around 0.2. There are no signs of significant overfitting as the validation loss aligns closely with the training loss. Furthermore, the accuracy plot shows consistent improvement, with training and validation accuracies reaching above 90% and stabilizing after epoch 50. The overlap between training and validation accuracy suggests generalization to unseen data. This translates to an accuracy of 0.95.

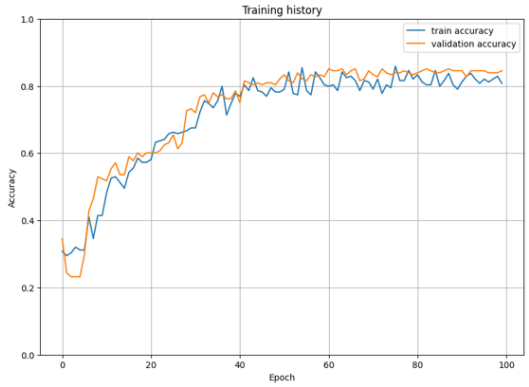

Figure 4.46: Accuracy Curve

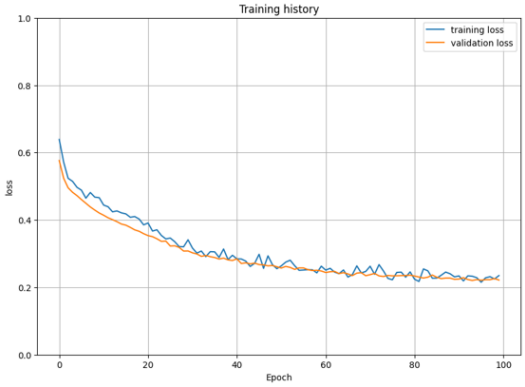

Figure 4.47: Loss Curve

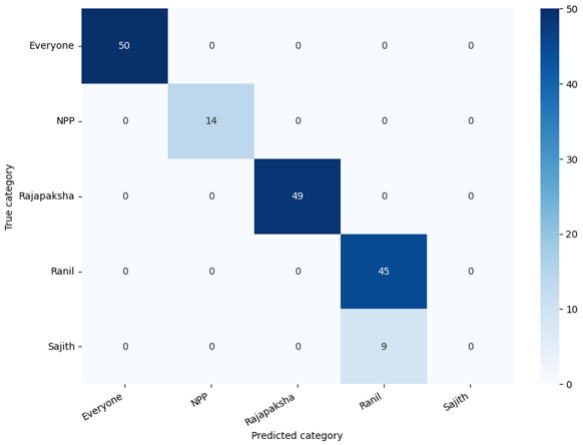

Figure 4.48: Confusion Matrix for BERT

Table 4.1 shows a comparison of the results obtained.

Table 4.1: Model Accuracy Summary

Model	**Model 1**	**Model 2**	**Model 3**
XGBoost	0.79	0.65	0.64
SVM	0.78	0.64	0.56
Random Forest	0.81	0.71	0.60
Naïve Bayes	0.75	0.70	0.56
sinBERT	0.82	0.76	0.95
AshenBERTo	0.94	0.77	0.71

The classification tasks in this study were distributed among the best-performing models based on their performance. For the task of identifying whether the content is Political or Non-Political (Model 1), AshenBERTo was selected as it demonstrated the highest accuracy (94%). The task of classifying sentiment as Positive or Negative (Model 2) was assigned to sinBERT, which achieved an accuracy of 76%. Finally, the task of detecting the Political Party affiliation (Model 3) was also handled by sinBERT, owing to its outstanding performance with an accuracy of 95%.

The selected models were then applied to the newly extracted data set. The new dataset has 25,336 entries. Table 4.2 shows the Positive and Negative comment counts per each candidate as "supportive" and "Unsupportive" comments.

Candidate	Unsupportive Comments	Supportive Comments
Anura Kumara Dissanayake	2452	12871
Sajith Premadasa	1907	3121
Ranil Wickramasinghe	909	1565
Every Politician	735	138
Namal Rajapaksha	563	181
Other Parties	414	480
Total	6980	18356

Table 4.2: Distribution of Negative and Positive Comments for Candidates

The graph in 4.49 illustrates the distribution of comments categorized by different candidates. It is evident that the NPP has the highest number of comments, significantly outpacing all others with 15,323 comments. This distribution suggests that discussions about the NPP dominate the conversation, indicating a strong public interest or engagement in this category compared to others.

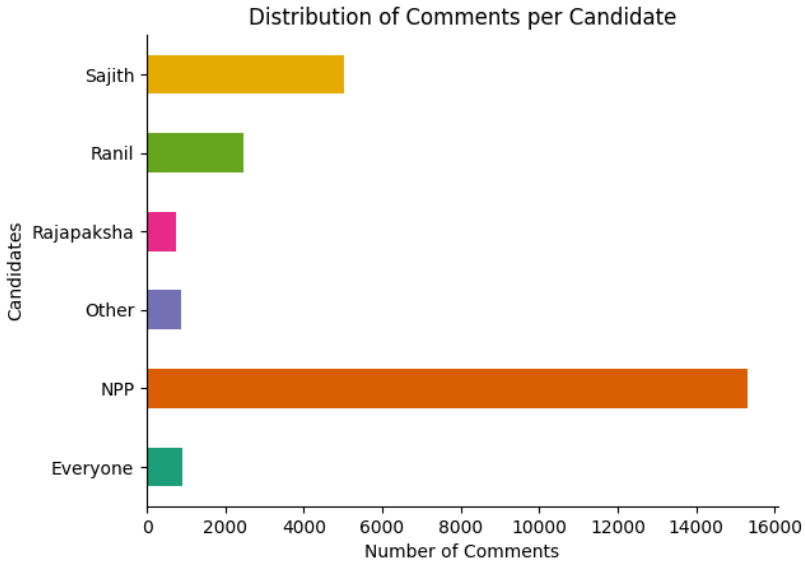

Figure 4.49: Total Comment Counts

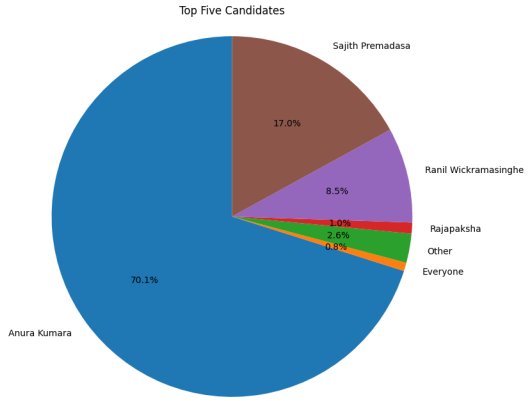

Figure 4.50: Supportive Comments Percentages

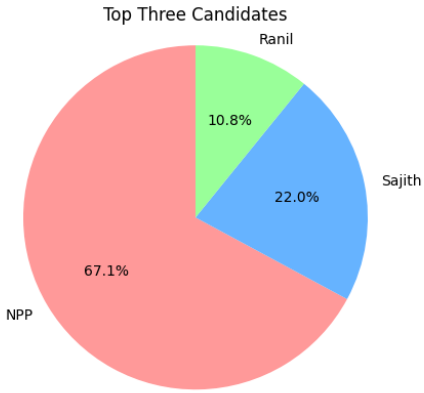

Figure 4.51: Supportive Comments Percentages

This pie chart in figure 4.50 depicts the distribution of supportive comments by political candidates. The majority of supportive comments, accounting for 70.1%, are directed toward Anura Kumara, indicating his significant popularity among the commenters. Sajith Premadasa follows with 17.0%, showing a moderate level of support. Ranil Wickremasinghe obtained 8.5% of the supportive comments, while Rajapaksha received a minimal share of 1.0%. The "Other" and "Everyone" categories account for 2.6% and 0.8%, respectively. These results highlight a dominant preference for Anura Kumara, with other candidates trailing significantly in terms of positive sentiment. Figure 4.51 shows the distribution of supportive comments per top 3 candidates.

According to the Department of Elections (2024), Anura Kumara Dissanayake secured first place with 42.31% of the votes, followed by Premadasa with 33.76%. The incumbent president, Wickremesinghe, finished third, receiving only 17.27% of the votes. Since no candidate received a majority, second-preference votes were counted. The following day, Dissanayake was declared the winner with 56% of the vote after these second preferences were considered. These results confirm that our model performs well in predicting the winning party, the first runner-up, and the second runner-up successfully.

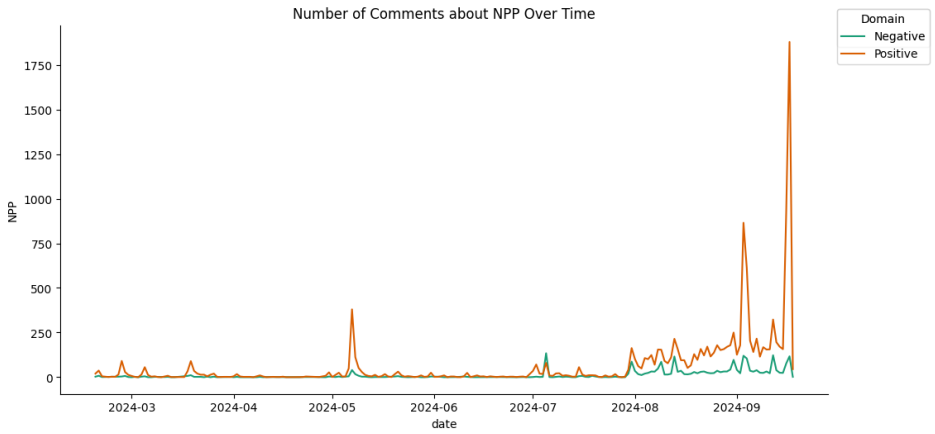

Figure 4.52: Comment Count over Time for NPP

This line chart in figure 4.52 shows the number of comments about NPP (National People's Power) and Anura Kumara Dissanayake over time, categorized into positive and negative sentiments. Over the period from March 2024 to September 2024, the volume of comments remained relatively low and stable, with occasional small spikes. However, starting around August 2024, there was a noticeable increase in both positive and negative comments. This trend escalated sharply toward September 2024, where positive comments experienced a significant peak, exceeding 1750 comments, while negative comments also increased but remained much lower in comparison. This suggests a growing engagement and positive sentiment toward NPP during this period, due to the political events or campaigns.

Name of Candidate	**Party Abbreviation**	**Votes Received**	**Percentage**
Anura Kumara Dissanayake	NPP	5,634,915	42.31%
Sajith Premadasa	SJB	4,363,035	32.76%
Ranil Wickremesinghe	IND16	2,299,767	17.27%
Namal Rajapaksa	SLPP	342,781	2.57%
Other			5.1%

Table 4.3: Election Results from Department of Elections (2024)

CHAPTER 5

CONCLUSION

This project aims to develop an AI-driven method to accurately predict election results using Natural Language Processing techniques. By fine-tuning two BERT base models, sinBERT and AshenBERTo, specifically designed for the Sinhala language and comparing their performance with machine learning classifiers, we identified the optimal strategy to achieve the above objectives. The integration of explainable AI techniques offered valuable insights into the decision-making process of these models by enhancing the interpretability of predictions. Additionally, the development of a method to predict election outcomes for Sri Lanka's presidential elections demonstrates the real-world applicability and impact of this research. The experiments show that the best performance was obtained by two BERT base models.

This work contributes significantly to the growing field of sentiment analysis in low-resource languages like Sinhala, providing a foundation for future advancements in political opinion mining and broader natural language processing applications.

REFERENCES

1. Atanasova, P. (2024). A diagnostic study of explainability techniques for text classification. In *Accountable and explainable methods for complex reasoning over text* (pp. 155–187). Springer.

2. Chathuranga, P., Lorensuhewa, S., & Kalyani, M. (2019). Sinhala sentiment analysis using corpus based sentiment lexicon. In *2019 19th international conference on advances in ict for emerging regions (icter)* (Vol. 250, pp. 1–7).

3. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321–357.

4. Demotte, P., Senevirathne, L., Karunanayake, B., Munasinghe, U., & Ranathunga, S. (2020). Sentiment analysis of sinhala news comments using sentence-state lstm networks. In *2020 moratuwa engineering research conference (mercon)* (pp. 283–288).

5. Department of Elections, S. L. (2024). *Presidential election results 2024*. Retrieved from `https://results.elections.gov.lk/pre2024/` (Accessed: January 9, 2025)

6. Dhananjaya, V., Demotte, P., Ranathunga, S., & Jayasena, S. (2022a). Bertifying sinhala–a comprehensive analysis of pre-trained language models for sinhala text classification. *arXiv preprint arXiv:2208.07864*.

7. Dhananjaya, V., Demotte, P., Ranathunga, S., & Jayasena, S. (2022b). Bertifying sinhala–a comprehensive analysis of pre-trained language models for sinhala text classification. *arXiv preprint arXiv:2208.07864*.

8. El Rifai, H., Al Qadi, L., & Elnagar, A. (2022). Arabic text classification: the need for multi-labeling systems. *Neural Computing and Applications*, *34*(2), 1135–1159.

9. Islam, M. Z., Liu, J., Li, J., Liu, L., & Kang, W. (2019). A semantics aware random forest for text classification. In *Proceedings of the 28th acm international conference on information and knowledge management* (pp. 1061–1070).

10. Jayasuriya, P., Ekanayake, S., Munasinghe, R., Kumarasinghe, B., Weerasinghe, I., & Thelijjagoda, S. (2020a). Sentiment classification of sinhala content in social media. In *2020 international research conference on smart computing and systems engineering (scse)* (pp. 136–141).

11. Jayasuriya, P., Ekanayake, S., Munasinghe, R., Kumarasinghe, B., Weerasinghe, I., & Thelijjagoda, S. (2020b). Sentiment classification of sinhala content in social media. In *2020 international research conference on smart computing and systems engineering (scse)* (pp. 136–141).

12. Kapoor, A., & Jindal, V. (2020). Exploring self organizing maps for brand oriented twitter sentiment analysis.
doi: 10.13140/RG.2.2.22212.45440

13. Mehta, H., & Passi, K. (2022). Social media hate speech detection using explainable artificial intelligence (xai). *Algorithms*, *15*(8), 291.

14. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 1135–1144).

15. Rish, I., et al. (2001). An empirical study of the naive bayes classifier. In *Ijcai 2001 workshop on empirical methods in artificial intelligence* (Vol. 3, pp. 41–46).

16. Syahrani, I. M. (2019). *Analisis perbandingan teknik ensemble secara boosting (xgboost) dan bagging (random forest) pada klasifikasi kategori sambatan sekuens dna* (Unpublished doctoral dissertation). Bogor Agricultural University (IPB).

17. Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.

18. Wang, H., Zhou, C., & Li, L. (2019). Design and application of a text clustering algorithm based on parallelized k-means clustering. *Revue d'Intelligence Artificielle*, *33*(6).

19. Xu, F., Uszkoreit, H., Du, Y., Fan, W., Zhao, D., & Zhu, J. (2019). Explainable ai: A brief survey on history, research areas, approaches and challenges. In *Natural language processing and chinese computing: 8th ccf international conference, nlpcc 2019, dunhuang, china, october 9–14, 2019, proceedings, part ii 8* (pp. 563–574).