

Detecting Propagandistic Poster Title:
A Machine Learning Approach

by

Riaz Mahmood
19201007

Intiajul Alam Shah
19301185

Tasnimul Hassan
19341001

Hasan Abdullah
19301247

Taskin Mohammad Mubassir
19201114

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
School of Data and Sciences, Brac University
March 2024

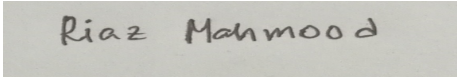
© 2024. Brac University
All rights reserved.

Declaration

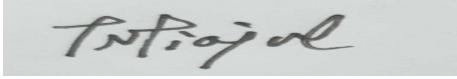
It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

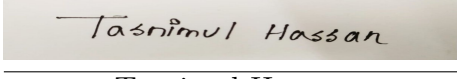
Student's Full Name & Signature:



Riaz Mahmood
19201007



Md Intiajul Alam Shah
19301185



Tasnimul Hassan
19341001



Hasan Abdullah
19301247



Taskin Mohammad Mubassir
19201114

Approval

The thesis titled “Detecting Propagandistic Poster Title: A Machine Learning Approach” submitted by

1. Riaz Mahmood (19201007)
2. Intiajul Alam Shah (19301185)
3. Tasnimul Hassan (19341001)
4. Hasan Abdullah (19301247)
5. Taskin Mohammad Mubassir (19201114)

Of Spring, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on March 10, 2024.

Examining Committee:

Supervisor:
(Member)



Dr. Md. Golam Rabiul Alam
Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi
Associate Professor; Chairperson
Department of Computer Science and Engineering
Brac University

Abstract

Detecting propagandistic content is crucial in today's digital age where misinformation spreads rapidly. In this study, we propose a machine learning approach aimed at identifying propaganda in poster titles. Our methodology encompasses various text classification techniques, including Random Forest, Logistic Regression, K-Nearest Neighbor (KNN), Naive Bayes classifier, Support Vector Machine (SVM), RoBERTa, Stacking Classifier, Stacking Classifier With Feature Engineering, and RoBERTa XGBoost Hybrid Model. We employ robust feature extraction methods such as TF-IDF and Word2Vec, along with advanced ensemble learning strategies, to enhance the accuracy and effectiveness of the classification process. Specifically, we introduce two hybrid models: the Stacking Classifier With Feature Engineering, which incorporates word2vec and TF-IDF to improve accuracy, and the RoBERTa XGBoost Hybrid Model, which utilizes a combination of TF-IDF vectorization and RoBERTa embeddings followed by XGBoost classification. Through extensive experimentation and evaluation, we analyze the performance of each model in terms of accuracy, precision, recall, and F1-score. Our findings demonstrate promising results, with certain models exhibiting significant improvements over baseline approaches. Moreover, we conduct a thorough analysis of the models' strengths and weaknesses, providing insights into their efficacy in detecting propagandistic content. Overall, our research contributes to the development of effective tools for combating propagandistic title and promoting media literacy in the digital landscape.

Keywords: Misinformation, Propaganda Identification, Visual Propaganda, Machine Learning Models, Societal Peacekeeping

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.

Secondly, to our advisor Mr. Golam Rabiul Alam sir for his kind support and advice in our work. He helped us whenever we needed help.

And finally to our parents without their throughout sup-port it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

Declaration	i
Approval	ii
Abstract	iv
Acknowledgment	v
Table of Contents	vi
List of Figures	viii
List of Tables	1
1 Introduction	2
1.1 Research Problem	3
1.2 Research Objectives	4
1.3 Research Challenges	4
2 Related Works	5
3 Methodology	8
3.1 Data Collection	9
3.2 Data pre-processing	9
3.2.1 Data Cleaning	9
3.2.2 Lower case	10
3.2.3 Tokenization	10
3.2.4 Removing special characters	10
3.2.5 Removing stop words and punctuation	10
3.2.6 Stemming	10
3.3 Feature Extraction	11
3.3.1 TF-IDF	11
3.3.2 Word2Vec	11
3.4 Description of Data	11
3.5 Data Preparation	12
3.6 Text Classification Techniques	14
3.6.1 Random Forest	15
3.6.2 Logistic Regression	15
3.6.3 K - Nearest Neighbor (KNN)	16
3.6.4 Naive Bayes classifier	16

3.6.5	Support Vector Machine (SVM)	16
3.6.6	RoBERTa	17
3.6.7	Stacking Classifier	18
3.6.8	Stacking Classifier With Feature Engineering	19
3.6.9	XGBoost Hybrid Model	21
4	Result and Discussion	23
4.1	Training Performance	23
4.2	Machine Learning Models	25
5	Conclusion	30
5.1	Conclusion	30
5.2	Future work	30
	Bibliography	32

List of Figures

3.1	Top level overview of the proposed propaganda detection system . . .	8
3.2	Flow Chart of the Data Collection Method.	9
3.3	Pie graph	12
3.4	Histogram graph of characters count	12
3.5	Histogram graph of words count	13
3.6	Class Distribution of the Dataset	13
3.7	Distribution of Token Length	14
3.8	Word Cloud for Propaganda and Non-Propaganda Text	14
3.9	Random Forest principle	15
3.10	Dynamic masking of RoBERTa	17
3.11	Byte Pair Encoding(BPE) using RoBERTa Tokenizer	18
3.12	Work process of Stacking Classifier model	19
3.13	Work process of Stacking Classifier model	20
3.14	XGBoost hybrid model	22
4.1	Classifier performance comparison before feature engineering	24
4.2	Classifier performance comparison after feature engineering	25
4.3	Naive Bayes Confusion Matrix	26
4.4	SVM Confusion Matrix	26
4.5	KNN Confusion Matrix	26
4.6	Logistic Regression Confusion Matrix	27
4.7	Random Forest Confusion Matrix	27
4.8	Stacking Classifier Confusion Matrix (Word2Vec + Tf-idf)	28
4.9	Stacking Classifier Confusion Matrix	28
4.10	RoBERTa Confusion Matrix	28

List of Tables

3.1	After data pre-processing ('Transformed Text')	10
4.1	Accuracy table of Machine Learning Models	23
4.2	Confusion matrix of Machine Learning Models	29

Chapter 1

Introduction

The most extensively acknowledged definition of propaganda was standardized by the Institute for Propaganda Analysis[1] and portrays the incident as actions performed by individuals or groups with the intention of persuasion of the opinion of target individuals. Propaganda has been used since 300 BCE. In the early stage, it was used to promote religious beliefs and political ideologies. Propaganda grew comprehensively during the 19th century. The reason behind it was the increased percentage of literacy that led propagandistic facts to reach wider audiences. Improved printing technology also helped to spread propaganda, as the propagandistic news, posters, etc were going into the hands of people cheaply and quickly. During World War I and World War II, propaganda was being used as a weapon which helped in spreading the news of fake war efforts, gaining public support and benefiting their agenda. Propagandistic posters have been used throughout history to make an impact on human life. The earliest example of propaganda posters is the art in the walls of ancient China and Egypt, where those arts tried to influence people's religious, political, and economic beliefs. During World War I "Uncle Sam Wants You" posters were very impactful according to the United States Of America's authority. A sufficient description of propaganda and the media used to spread it in this day and age is provided in the introduction [2]. These days, the practice of democracy in the vast majority of the countries of the world has given the people freedom of speech. This freedom is misused by some particular groups or some people who have a motive to circulate some information that is not even valid and can create hatred towards any person or group of people. Also, the availability of the internet has opened the door of exploration to people. In general, we are living in an era of "Information and Technology", which is making life easier. Large-scale digital content is being produced because of arbitrary access to the World Wide Web. In particular, the expansion of Internet-based communication has turned into a global necessity. Despite these benefits, a lot of individuals nevertheless try to utilize the Internet in negative ways. Both freedom of speech and the availability of the internet are heaven for the ones who are trying to manipulate or spread information that is not valid or has a purpose behind it. This misinformation sometimes expands through posters that can be related to politics, movies, or valid or invalid protests to establish any agenda. Propaganda can remain hidden in those posters which can be of a movie or of an ongoing protest, or the posters we see on the side-walls of the sidewalks. Due to the large number of internet users who are connected to social media, it is pretty simple for any kind of information to spread quickly

and affect the user's view. From an authenticity standpoint, this type of material, commonly known as viral content, is frequently unverified and can cause outrage or frenzy. According to the FBI's Internet Crime Complaint Center, more than 791,000 complaints about criminal conduct that used the internet as a facilitator were received in 2020 which was almost double than 2019 and the expected loss was \$4.2 billion [13]. Propaganda is an act done by an opportunistic side for their benefit or to harm another party. It directly aims towards establishing an agenda. Propaganda remains hidden in news outlets that can be well known or not and have the ability to reach a vast audience [3]. An inexperienced reader would never find out and believe those invalid pieces of information. That's where propaganda gets successful. Propaganda motivates people towards criminal activities and to have a particular perspective. Also, it can downgrade anyone which leads someone to disaster. To legitimize or train the public in illegal operations, terrorist organizations, to manipulate votes or religious sentiments, propagandist posters are often used. That is why the people who want to spread propaganda utilize the Internet, the sidewalks and the open-spaces as an active means to put posters where there is propaganda inside. We aim to detect those propaganda before reaching the general people. If we identify and alert the viewers in time we will be able to avoid those problems regarding propaganda.

1.1 Research Problem

The world wide web, blogs, social media, forums, and other online platforms have been delivering a significant amount of digital media content over the past few years as the Internet becomes more convenient and effortless. It has been researched that among all the contents there are some fake, fabricated, and propagandistic as well as genuine and useful. Before this digital era, news and information used to circulate through paper media which was mostly in newspapers, magazines, and poster prints. In the 2016, US presidential election, the news media aimed to influence the election and falsify people about the true situation [5]. As we see news, articles and posters influence people to have a mindset of particular perspectives, It is a matter of concern that it should be done in the proper way. Because of this reason, we intend to focus on this field and aim to bring out the best result to detect propagandistic posters of different events and time. Posters were used as a main tool for influencing mass people as there was no such digital media [10]. A large amount of research has been done about detecting the propaganda in news articles and social media content. However, there are no significant works on detecting propagandas in posters and that like art. So, it will be challenging for us to find enough resources and datasets for testing and training. However, to make an impact on this field we have to work on developing the system. There has been some research about identifying propagandas from news articles, social media posts, blogs and documentaries using machine learning models. Since almost every one of them is in a different category we feel the urge to do this research in our own motifs. We intend to create our own dataset and will keep it open and continuously updated for future uses.

1.2 Research Objectives

Propaganda is an act done by an opportunistic side for their own benefit or to harm another party. This research aims to develop a System for detecting propaganda from poster titles using Machine learning models such as Logistic Regression (LR), Random Forest (RF) and the KNN. The objectives of this research are:

1. To understand the machine learning (ML) models and how it works.
2. To detect propagandistic information from poster texts and also to select non propagandistic information.
3. To develop a model for a propaganda detection system based on machine learning.
4. To evaluate different Machine Learning models.
5. To offer recommendations on improving the models.

1.3 Research Challenges

As we proceeded more into the research work, we found several challenges to work on. We did not find any benchmark work on this specific title category and we collected the dataset manually. While collecting the dataset from different sources we had to recheck the authenticity because propaganda itself has different aspects. While extracting the titles of posters many irrelevant data was collected which we needed to remove and rearrange the preferred data.

After that, we faced challenges comparing our model results because of the lack of benchmark works. We can not be assured of our proposed model whether will be effective or not effective. There are some other issues we would like to mention in the following points:

1. The scarcity of labeled data for propaganda detection, especially for low-resource languages and diverse domains. This limits the generalization and robustness of our machine learning model, as well as the evaluation of its performance.
2. Increasing the training data will increase the computation time and the training time unless we use a specific GPU such as T4, which is designed for computational tasks, but is expensive and only accessible in cloud service platforms.

Chapter 2

Related Works

Propaganda, a willful propagation of information, facts, rumours, fake and groundless news or lies to make an impact on public view or opinion which always has an aim behind it. This part focuses on the attempts that has been made in the propaganda detection field. In order to produce better results, every new scientific discovery is built on prior research and then modified. The number of new technology, social media, online newspaper is growing rapidly. Because of that these sources produce information which are based on current events, therefore it's very essential to identify which ones are real as well as which ones are fake. As a result, it is difficult for the research community to manually recognise such papers. They suggested a number of fresh methods for the automated identification of propaganda from online web sites. The concept of text classification was initially proposed in the 18th century, but as classification techniques advanced, academics started categorising texts into other groups, such as news, web pages, etc. In the past, researchers have proposed a number of methods for spotting propaganda and fraudulent information from online web sources. Every study project's core foundation is the investigation of stylometric, writing, or readability aspects. We read some publications which used various models, including BERT, BiLSTM, CNN, and LSTM-CRF. Recent interest has surged in deciphering methods for detecting and classifying textual propaganda. In these collection of works Barron-Cede'na et al. [2019] [3], proposed a technique to categorise news stories according to the amount of propaganda content present in each article and develop a fresh dataset (QProp), which has been meticulously labelled concerning propaganda versus nonpropaganda content.

Providing comprehensive information regarding the creation of each news item, reliable classifications Barron-Cede'na et al. [2019][4] proposed a new approach for detecting propaganda in news articles by analysing text fragments rather than labelling entire documents or news outlets. The authors argued that previous methods suffered from noisy labels and lack of explainability. They introduced a corpus of manually annotated articles, focusing on specific propaganda techniques at the fragment level. The article describes eighteen propaganda techniques, such as loaded language, repetition, appeal to fear, and obfuscation etc. They provided examples for each technique. The authors also presented a novel multi-granularity neural network that outperforms existing models based on BERT. They emphasised the importance of high-quality professional annotations and released their corpus and code for future research. Overall, the article contributes to fine-grained analysis of propaganda techniques in texts and provides a curated evaluation framework for

studying propaganda.

Another article by Vlad et al.[2019][9] presented a comprehensive study on sentence-level propaganda detection in news articles, focusing on the development of a robust binary classifier using transfer learning and a unified neural network model called BERT-BiLSTM-Capsule. Propaganda, a powerful tool throughout history for influencing public opinion, has gained new dimensions with the advent of online social media platforms. The authors participated in the NLP4IF-2019 Shared Task SLC, where they ranked 12th among 26 teams, achieving promising results with an F1-score of 0.5868. The proposed model combines BERT for word encodings, BiLSTM for capturing semantic features, and Capsule Networks for selecting salient features, ultimately improving upon the baseline approach of the task organisers. Additionally, the authors explored the relationship between emotions and propaganda, employing a pre-training approach on an emotion-labelled dataset to enhance the performance of their system. The study highlighted the potential of the BERT-BiLSTM-Capsule model in propaganda detection and suggested further exploration with contextualised embeddings like ELMo and FLAIR. Overall, the authors' system, BERT-Emotion, showcases significant advancements in fine-grained propaganda detection and opens avenues for future research in this domain.

In their paper Gupta et al.[2019] [6] introduced MIC-CIS, a comprehensive system that skillfully detects fine-grained propaganda in news articles by leveraging a diverse range of neural architectures such as CNN, LSTM-CRF, and BERT, in combination with linguistic, layout, and topical features, enabling efficient handling of both sentence-level and fragment-level propaganda detection tasks. Through the use of multi-granularity and multi-tasking neural architectures, the system achieved competitive performance in both tasks. Furthermore, ensemble strategies, such as majority-voting and relax-voting, were explored to enhance the overall system effectiveness. The authors' submissions ranked 3rd and 4th among the participating systems in fragment-level and sentence-level propaganda detection, respectively. The study highlights the importance of addressing propaganda detection at fine-grained levels and promotes the development of explainable AI. The authors suggested future work involving the incorporation of linguistic, layout, and topical features into the fine-tuning process of BERT models, as well as exploring methods to extract salient fragments for better understanding of neural network learning and promoting explainable AI.

An online prototype called Prta that was trained on misinformation articles was recently proposed by Martino et al.[2020] [15] . In this sample, users may enter plain text or a URL, but they cannot download the results. Similar to PROTECT, Prta analyses the use of propaganda tactics on predetermined subjects and displays the propagandist messages at the snippet level with the opportunity to filter the propaganda techniques to be presented depending on the confidence rate. This system's implementation is based on the methodology suggested in Da San Martino et al., [4].

Later in their research Martino et al. [2020] [12] presented 'The Detection of Propaganda Techniques in News Articles', addressing the challenge of identifying and classifying propaganda techniques used in text. Propaganda, with its influential nature and potential reach, poses a significant threat in today's media landscape. The task focuses on fine-grained analysis, aiming to develop models capable of detecting specific text fragments containing propaganda techniques. The article describes

the task organisation, corpus, and evaluation metrics, highlighting the participation and results of the participating teams. Fourteen curated propaganda techniques are identified, such as loaded language, repetition, appeal to fear, and straw man, among others. The task proves to be challenging, particularly in technique classification, emphasising the need for further research and expanded datasets. The authors acknowledged the ethical considerations of deploying automatic propaganda detection systems and emphasise the importance of raising awareness and empowering users to discern propaganda independently.

Yu et al.[2021][16] presented a novel approach for detecting and interpreting propaganda in news articles, addressing the need for accurate and interpretable systems to combat misleading content. By analysing qualitative descriptive features and their suitability in detecting deception techniques, the proposed system offers interpretability by showcasing the specific techniques used in propagandistic content. Additionally, the article highlights the importance of interpretability in building trust and acceptance among users. The system combines interpretable features with pre-trained language models, achieving state-of-the-art results. Furthermore, the authors emphasised the rise of citizen journalism and the challenges of fact-checking and identifying bias and propaganda in online media. The proposed model not only detects propaganda but also explains its predictions, aiding users in understanding why certain content is deemed propagandistic. The article concludes with future plans to expand the dataset and release an interpretable online system to foster a healthier and safer online news environment.

Investigating additional novel approaches to identify propaganda by employing language models and leveraging transformer-based frameworks was an examination part of Mapes et al.[2019] [8]. Using a refined BERT structure combined with parameter tuning, the winning approach on NLP4IF'19 maximised sentence-level classification accuracy .

Following a thorough examination of pre-processing techniques, Yoosuf and Yang [2019] [11] shifted their focus toward highlighting specific aspects of language models and contemporary propaganda tactics before applying the BERT framework to reframe the problem by recognizing it as a sequential labelling concern. The systems that took part in the SemEval 2020 Challenge - Task 11 represent the most recent approaches to identify propaganda techniques based on given propagandist spans. The most interesting and successful approach by Jurkiewicz et al.[2020] [14] proposes first to extend the training data from a free text corpus as a silver dataset, and second, an ensemble model that exploits both the gold and silver datasets during the training steps to achieve the highest scores.

Chapter 3

Methodology

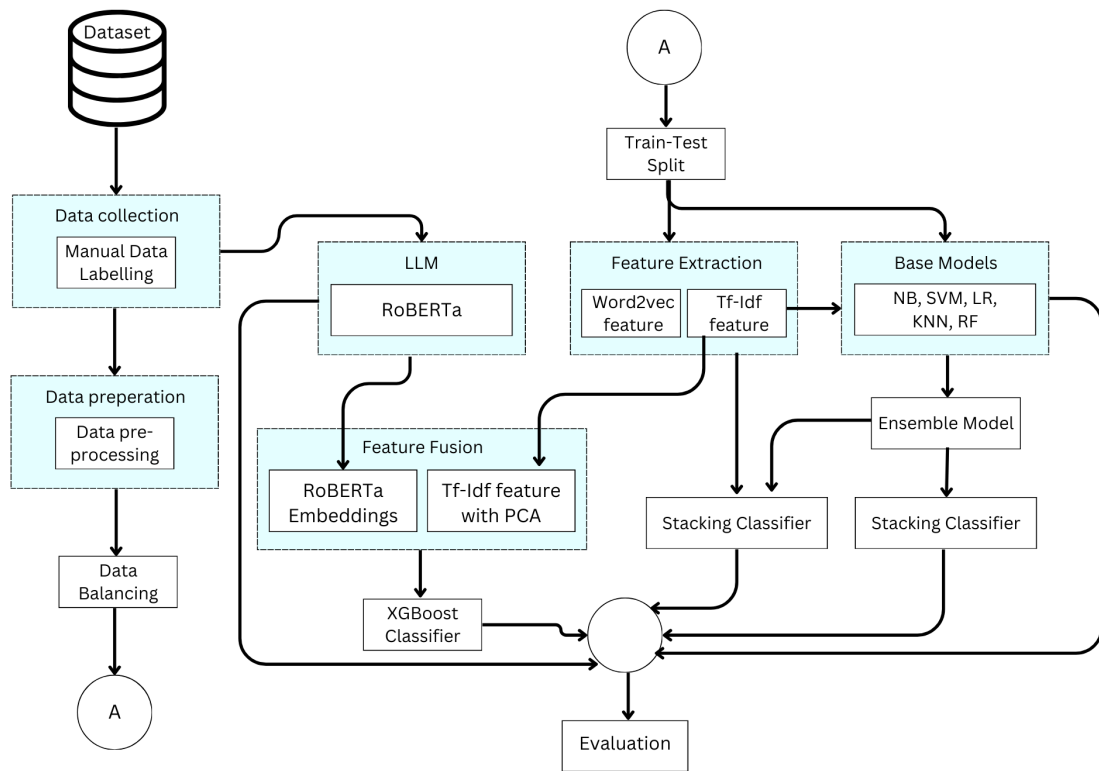


Figure 3.1: Top level overview of the proposed propaganda detection system

The Methodology section outlines the research strategy, data collecting, data preparation, and classification techniques employed to construct the machine learning-based system for identifying propaganda in poster titles. Furthermore, it elucidates the reasoning behind selecting these methodologies and how they correspond to the study inquiries and goals. The Methodology section elucidates the reader regarding the extent, constraints, and soundness of the study. In Figure 3.1, the steps of our proposed system are illustrated.

3.1 Data Collection

Given the limited availability of research on our subject, we gathered data from multiple websites that featured titles of both propaganda and non-propaganda posters. We concentrated on posters originating from significant historical events such as World War I and II, and corroborated their genuineness through dependable sources. In addition, we gathered non-propagandistic titles from movie and documentary posters, specifically excluding those that contained any propaganda components in their content. We employed two techniques for data scraping and acquired approximately 600 titles classified as propaganda and 3000 titles classified as non-propaganda. Subsequently, we conducted data preprocessing by eliminating duplicate entries, texts in other languages, and extraneous content. In Figure 3.2, we illustrated the data collection method and mentioned the tool We used for the primary data scraping.

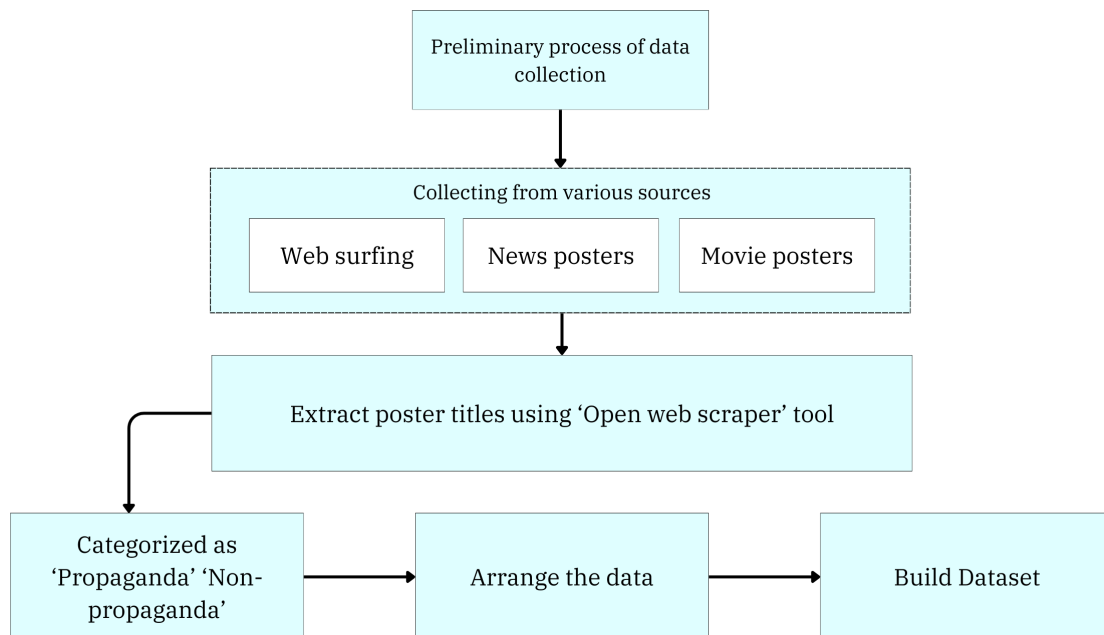


Figure 3.2: Flow Chart of the Data Collection Method.

3.2 Data pre-processing

3.2.1 Data Cleaning

To find and correct flaws in the dataset, such as missing values or duplicate entries, we first concentrate on data cleaning. Clean data can assist increase the accuracy of analysis or models and eliminate future devastating mistakes. We initially had four columns and 3055 rows with no data in the 'Unnamed: 0' column, and we did not require the 'Source' column for our tasks. We eventually tested the dataset's null value after dropping these columns and all of the other null values. Null values are absent from our dataset. However, there were 184 duplicate records, so we removed

them. We can now work with data that is free of duplicates and empty fields. After data cleaning, it changes from 3055 data to 2870 data.

3.2.2 Lower case

Lowercasing is the technique of making all the letters in a string of text lowercase. This maintains consistency across the text and makes it simpler to read. For example, “Aria Of A Starless Night” become “aria of a starless night”. All the words in lowercase.

3.2.3 Tokenization

Another method for breaking down a text into smaller, analytically-relevant pieces is tokenization. We get the words in this process. For example, “aria of a starless night” becomes [‘aria’, ‘of’, ‘a’, ‘starless’, ‘night’].

3.2.4 Removing special characters

After that, the text has to be free of special characters like punctuation, hashtags, and user mentions. For example, “aria of a starless night” Stop words like ”and” or ”the” can be removed because they do not help readers comprehend the context very well.

3.2.5 Removing stop words and punctuation

Stop words like ”and” or ”the” can be removed because they don’t help readers comprehend the context very well.

3.2.6 Stemming

Last but not least, stemming breaks words down to their basic components, treating variants like ”loving,” ”loved,” and ”loves” as a single word. We create “transformed text” using all the processes.

Propaganda text	Type	num of character	num of words	num of sentences	transformed text
Sword Art Online: Progressive - Aria of A Star...	0	56	11	1	sword art online progress aria starless night
The Personal History Of David Copperfield	0	41	6	1	personal history david copperfield

Table 3.1: After data pre-processing (‘Transformed Text’)

3.3 Feature Extraction

ML algorithms are unable to understand from the texts we have provided. Feature extraction basically uses mapping to identify sense in these texts.

3.3.1 TF-IDF

As it is based on word statistics, TF-IDF that is Term Frequency – Inverse Document Frequency, has been used for machine learning models to extract features from texts. Here two statistical techniques are used by TF-IDF: Term Frequency and Inverse Document Frequency. TF-IDF vectorization, a document term matrix is produced, with weight—a metric indicating a word’s significance for a particular text message—in the cells and specific, unique terms like ”count vectorizer” in the columns. Weight calculation formula is,

$$W_{x,y} = tf_{x,y} \cdot \log \left(\frac{N}{df_x} \right)$$

In this case, N is the total number of documents in the corpus, is the number of documents containing the term x, and term frequency, or tf_{xy} , is the number of times term x occurs in y divided by the total number of terms in y. A single propaganda or non-propaganda title in the present case is a document. Therefore, the inverse document frequency serves as a measure for the quantity of information a word contains. For example, we take a propaganda/non- propaganda title, $y =$ ‘American Prisoners of War Cared for by the Red Cross’ from our dataset. Let, term, $x =$ red So, term frequency, $tf_{red,y} = 1/9 = 0.11$. As dataset contains 2870 title total number of documents, $N = 2870$, $df_{red} = 39$ as the word ‘red’ is present in 39 titles in dataset. So, weight of the word,

$$W_{red.x} = tf_{red.y} \cdot \log \left(\frac{2870}{39} \right) = 0.2$$

We applied TF-IDF Vectorization on our datasets after data pre-processing.

3.3.2 Word2Vec

Word2Vec is a procedure of learning word embeddings where transforming texts into vectors. There are two strategies for training Word2Vec: Continuous Bag of Words and Skip-Gram.

Continuous Bag of Words is increasing the context by using the surrounding words to predict what occurs in the middle. For example, ‘Troll2 is great’ here, using Troll2 and great, predict the word between them, ‘is’. Skip-Gram is increasing the context by using the word in the middle to predict the surrounding words. For example, ‘Troll2 is great’ here, using ‘is’, predict the surrounding words Troll2 and great.

3.4 Description of Data

We can observe from the bar graph in figure 3.6 that there are more than 2481 non-propaganda data points and almost 575 propaganda data points. The pie graph

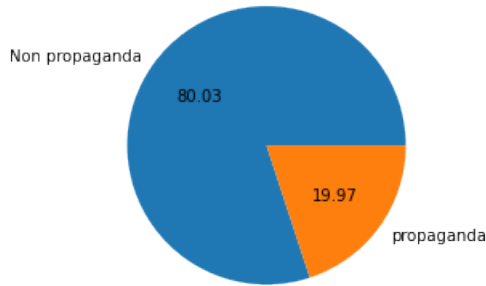


Figure 3.3: Pie graph

in figure 3.3 shows that 19.97% of this info is propaganda and 80.03 percent is non-propaganda. As a result, we might conclude that the data are unbalanced. For non-propaganda, the maximum and minimum character counts are 97 and 2, respectively. Contrarily, for propaganda, the maximum and minimum character counts are respectively 232 and 4. On the first histogram graph figure 3.4, where

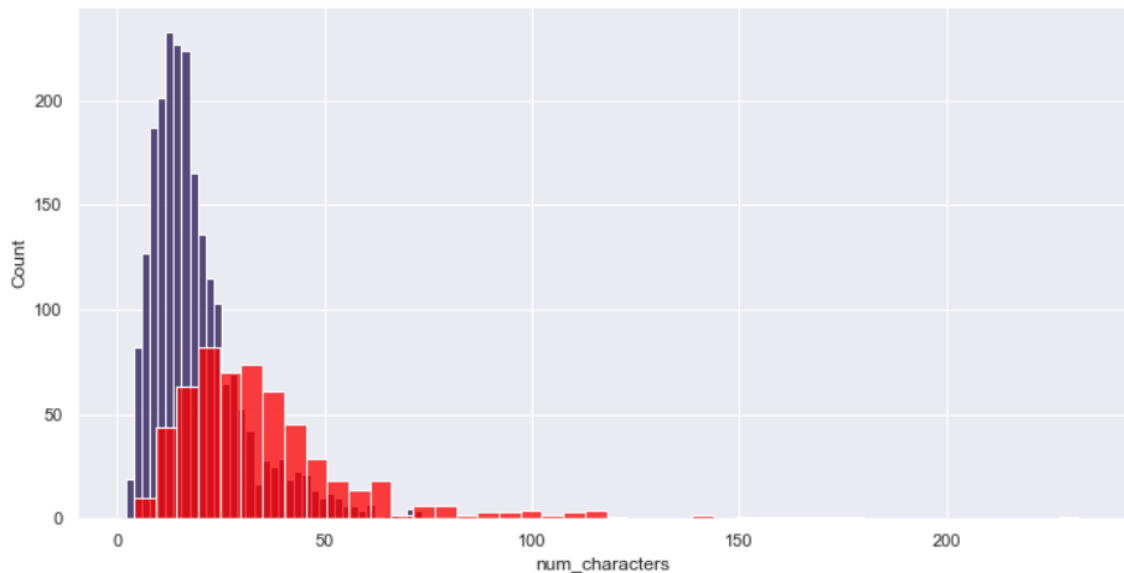


Figure 3.4: Histogram graph of characters count

the red color denotes non-propaganda and the purple color, propaganda, we can observe this character count. The second histogram graph figure 3.5 shows the difference between propaganda and non-propaganda in terms of words. So we might conclude that the number of characters and words in propaganda is higher than in non-propaganda, and mostly only one sentence is used for both.

3.5 Data Preparation

We imported the dataset file, which contained poster titles and their types (propaganda or non-propaganda). We removed irrelevant columns ('Unnamed: 0' and 'Source'), missing data, and duplicates. We then cleaned and normalized the text

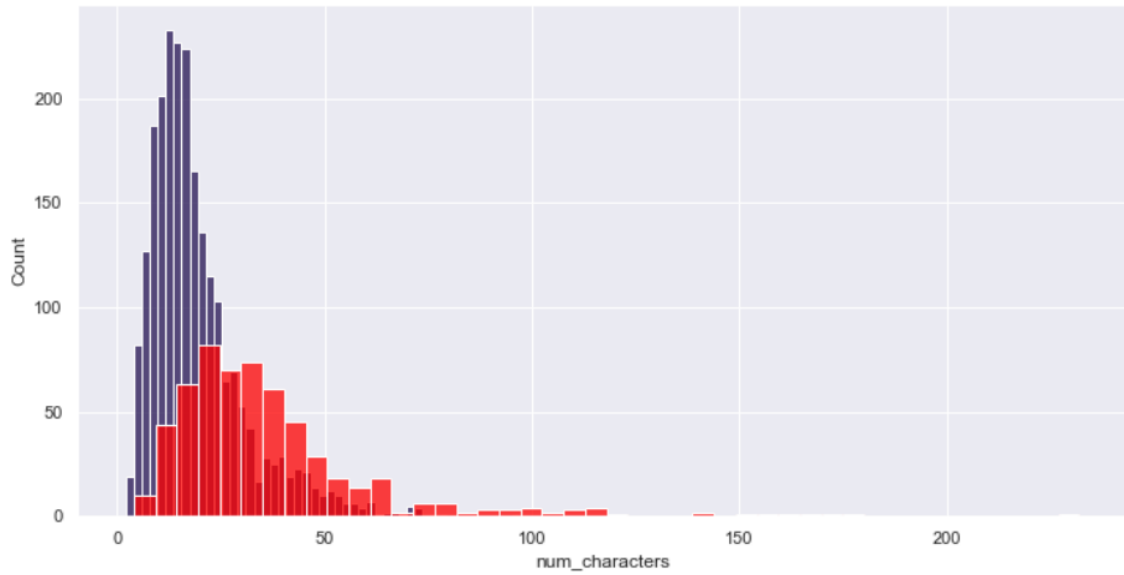


Figure 3.5: Histogram graph of words count

data by converting it to lowercase, removing punctuation, tokenizing words, eliminating stop words, and applying stemming. We used the word_tokenize function from the NLTK package, the English stopwords list, and the PorterStemmer algorithm. We also mapped the ‘Type’ labels to 0 (non-propaganda) and 1 (propaganda). We split the data into training 80% and testing 20% sets, using the train-test-split function from the sklearn library and setting the random state to 42.



Figure 3.6: Class Distribution of the Dataset

The bar graph depicted in Figure 3.6 illustrates the disparity in our dataset, with 2000 titles classified as non-propaganda and 500 titles classified as propaganda. Figure 3.7 shows the token length distribution in a dataset. The x-axis is “Token Length” (0-60) and the y-axis is “Count” (0-1000). Most tokens are 5-15 characters long, as shown by the histogram peak and curve. The dataset has mainly short and medium-length tokens, such as words, phrases, or sentences. The distribution has a long tail, indicating some very long tokens, such as paragraphs or documents. The distribution shape may reflect the data source and the tokenization method. Figure 3.8 shows the word frequency in propaganda and non-propaganda texts. The

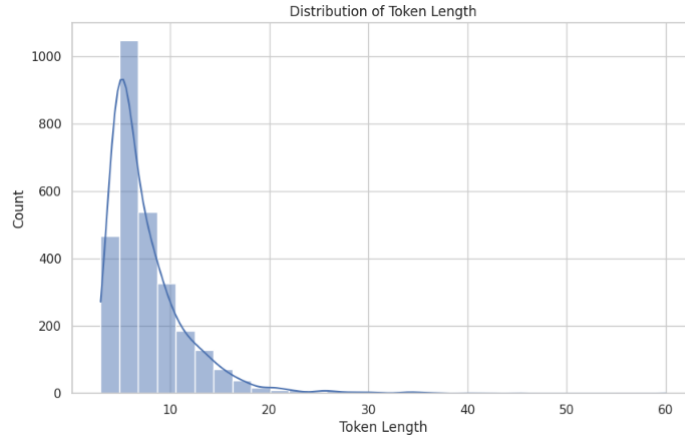


Figure 3.7: Distribution of Token Length

x-axis is “Word” and the y-axis is “Frequency”. Two word clouds are displayed side by side, labeled “Word Cloud for Propaganda Text” and “Word Cloud for Non-Propaganda Text”. The word size indicates the frequency; larger size means higher frequency. The propaganda text word cloud features words like “War”, “Freedom”, “American”, and “Country” prominently, indicating their frequent use in propaganda materials. In contrast, the non-propaganda text word cloud highlights words

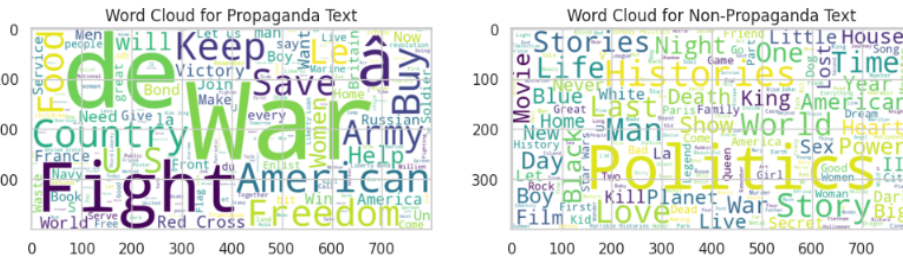


Figure 3.8: Word Cloud for Propaganda and Non-Propaganda Text

such as “Politics”, “World”, and “Man”. These visual representations serve as a comparative analysis tool to discern the lexical choices characteristic of both types of texts.

3.6 Text Classification Techniques

Text classification involves the assignment of a label to a text based on its content. It is extensively utilized in applications of natural language processing, including sentiment analysis, spam detection, topic modeling, and other related tasks. This work uses text classification to identify propaganda in poster titles, presenting a unique and demanding task. We employ a total of six distinct machine learning models for the purpose of text classification, each possessing its own unique set of benefits and drawbacks. The models include Random Forest, Logistic Regression, K-Nearest Neighbor (KNN), Naive Bayes Classifier, and Support Vector Machine (SVM). In the subsequent subsections, we provide a comprehensive overview of the primary characteristics and variables of each model. In addition, we assess the

models' performance by employing diverse assessment criteria, including accuracy, precision, recall, and F1 score.

3.6.1 Random Forest

The Random Forest is an ensemble classifier that leverages more than one learning algorithm to reap improved predictions in comparison to individual algorithms. This classifier includes a group of decision trees working in unison. The "n-estimator" option determines the number of decision trees included in the random forest. While a larger value for n-estimator may also result in slower performance, it yields greater accurate results.

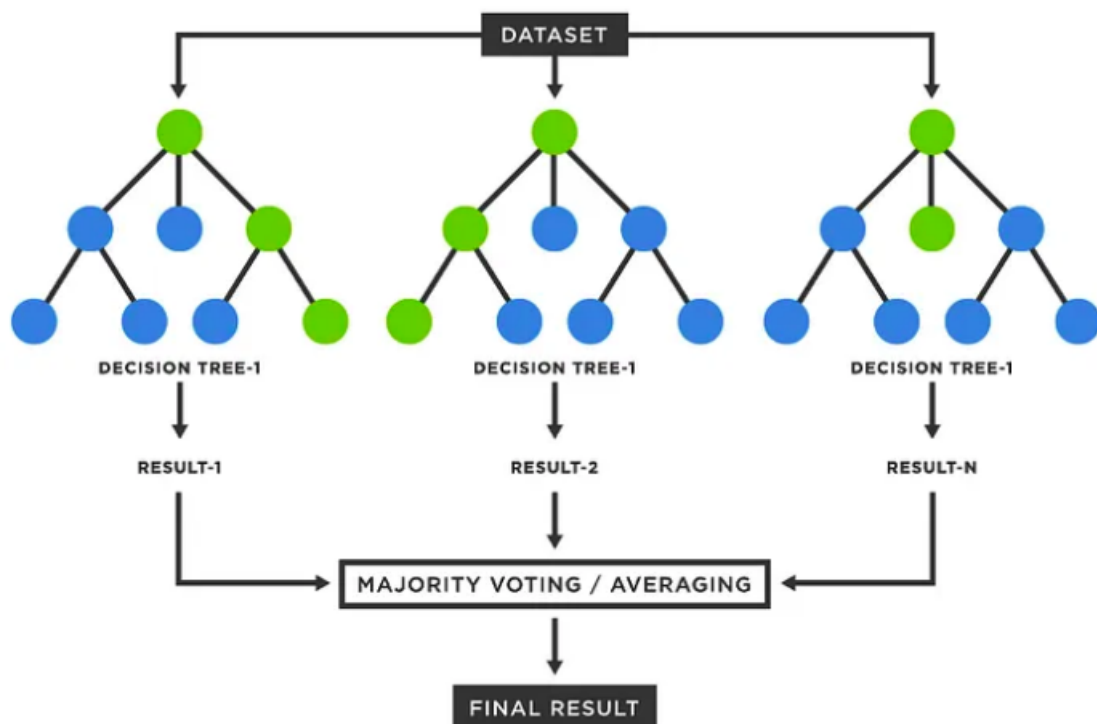


Figure 3.9: Random Forest principle

3.6.2 Logistic Regression

To decide the probability of an event taking place, the logistic regression classification method is employed. Typically, activities are characterised via a binary nature, with two potential values: 0 and 1. In this context, 0 indicates the absence of the event, even as 1 shows its occurrence. For example, logistic regression was applied on this task to forecast whether an individual had diagnosed propaganda via the application of machine learning techniques. If propaganda was detected, the corresponding column might be assigned a value of 1, whereas a value of 0 might be assigned to the non-propaganda column. There are three types of logistic regression: multinomial logistic regression, ordinal logistic regression, and binary logistic regression. Because logistic regression makes some assumptions and can predict whether a person can recognize propaganda or not, we used binary logistic regression in our model. It is assumed that the outcome will be binary, the dependent variable will

be categorical, the independent variable will not be multicollinear, and the sample sizes will be adequate when using logistic regression.

3.6.3 K - Nearest Neighbor (KNN)

K-Nearest Neighbor is a nonparametric classifier for supervised learning (KNN). This algorithm divides a single point into groups based on the separation between that point and its nearest neighbors. The number of neighboring points that must be expected or grouped together is denoted by the letter "K" in KNN. The KNN technique is useful for solving regression and classification issues. Our project is a classification challenge that asks us to identify propaganda, hence we choose KNN as a classification approach. In order to put KNN into practice, we must first select a value for K, where K is an integer that represents the number of neighbors we will take into account. Following the selection of K, we must determine the distance between a single point and K points using a variety of approaches, including the Euclidean distance, Manhattan distance, Hamming distance, etc. After calculating the distance between a single point and its neighbors by using one of the distance metrics listed above, we must select the K closest neighbors based on that distance. The next step is to count the number of data points among these K nearest neighbors before classifying a single data point into the category with the greatest number of nearest neighbors. Once we have categorized the new single data points into various categories, the KNN model will be prepared.

3.6.4 Naive Bayes classifier

The application of Bayes' theorem with strong independent estimates between features forms the basis of the family of simple "probabilistic classifiers" known in statistics as Naive Bayes classifiers. Although they are among the most straightforward Bayesian network models, when combined with kernel density estimation, they can produce results with high accuracy.

$$p(y|x) = \frac{p(x, y)}{p(x)} \quad (3.1)$$

Naive Bayes classifiers are highly scalable because the variety of parameters they need is linear in the set of variables in a machine learning model. Maximum Likelihood training can be carried out by analyzing a closed-form expression, which takes linear time, as opposed to applying an expensive incremental approximate solution, which is the number of other types of classifiers trained.

3.6.5 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised machine learning algorithm which is used to find the best possible way to separate different text data and solve classification and regression problems. In this algorithm, the data points are separated by a hyperplane and the hyperplane is determined by a kernel (Linear Kernel, Polynomial Kernel, Radial Basis Function Kernel). This kernel of a support vector machine is a special technique that can convert lower-dimensional data into a higher-dimensional space.

To identify propaganda and non-propaganda, this SVM technique learns from labeled examples of texts representing both categories. In this technique, it looks at several features within the text, like words, phrases, or sentence structures, and aims to create a boundary that effectively separates these categories. This hyperplane or boundary is like a line or plane that differentiates the texts into propaganda and non-propaganda areas in a high-dimensional space. Support Vector Machines goal is to find the hyperplane or boundary in such a way that it maximizes the space between different classes. When the model gets a new text, SVM makes use of this learned boundary to predict, using the patterns and traits it discovered from the labeled examples, whether the text falls into the propaganda or non-propaganda category based on where it falls to this boundary in the feature space. Support Vector Machine helps identify propaganda behavioral patterns in a range of textual datasets because of its resilience and adaptability.

3.6.6 RoBERTa

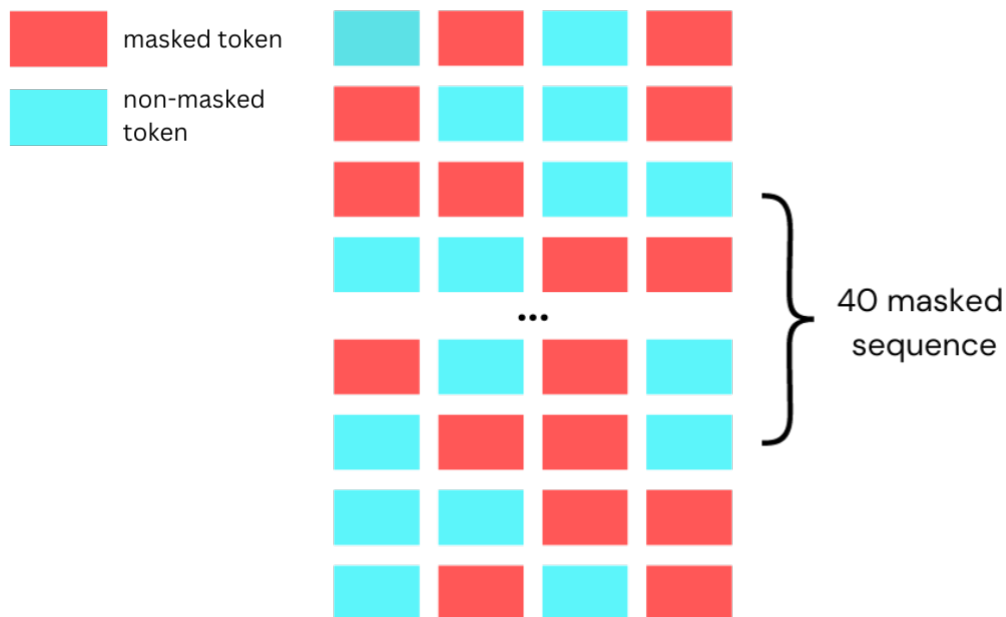


Figure 3.10: Dynamic masking of RoBERTa

The RoBERTa (Robustly optimized BERT Pre-training approach) model is a variant

of BERT that has been optimized and used for better performance. RoBERTa modifies some of the parameters of BERT and outperforms the score in several tasks such as GLUE and SQuAD. As opposed to the BERT's static masking, it uses dynamic masking where the masked tokens are changed in each epoch [7]. Our purposal is to use the model for sequence classification. We initialize the tokenizer

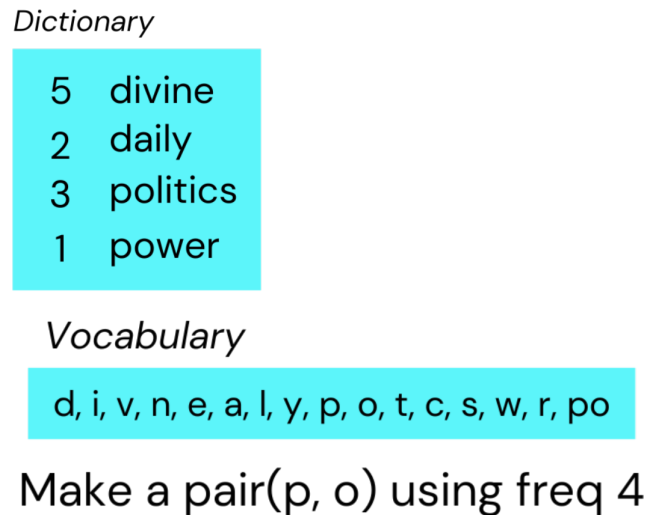


Figure 3.11: Byte Pair Encoding(BPE) using RoBERTa Tokenizer

using Roberta Tokenizer to encode the text into a format compatible with the model using Byte pair encoding (BPE) that it expects at the beginning and end of the sentences.

The sequence classification method is loaded with the pre-trained method. This method initializes the model with pre-trained weights which was declared with the roberta-base model. Then the model is trained using a custom training loop in five epochs. As it is pre-trained with larger batch sizes and has a reach vocabulary, the model helps us to leverage the text-based dataset and give us results in less time. We use this RoBERTa method as it is a state-of-the-art transformer model and outperforms the BERT and other models on a variety of NLP tasks.

3.6.7 Stacking Classifier

The research with which ensemble modeling is handled leads to the deployment of the Stacking Classifier for text classification tasks and the objective is to allow machines to sort the text into individual groups accurately. Ensemble modeling is a strong framework that works for the solution of simultaneously supplied data sets by different predictive models to get better forecasts. This goal is achieved through the consideration of different modeling algorithms and datasets having distinguishing characteristics, the ensemble model then synthesizes the projections from all the base models to yield a final and complete estimate of new data points. The stacking algorithm falls within the field of ensemble modeling methods and is one of many most powerful techniques. It can do this by combining the predictions of

different base models into an ensemble system which increases the overall accuracy, hence being given several names like stacked combinations, aggregations, or classifier stacking. These methods are Naive Bayes, Support Vector Machine (SVM), K Nearest Neighbors (KNN), and Random Forest to name a few. In particular, the combination of such models allows joint training to assess multiple text features simultaneously, improving the text classification’s overall quality and reliability.

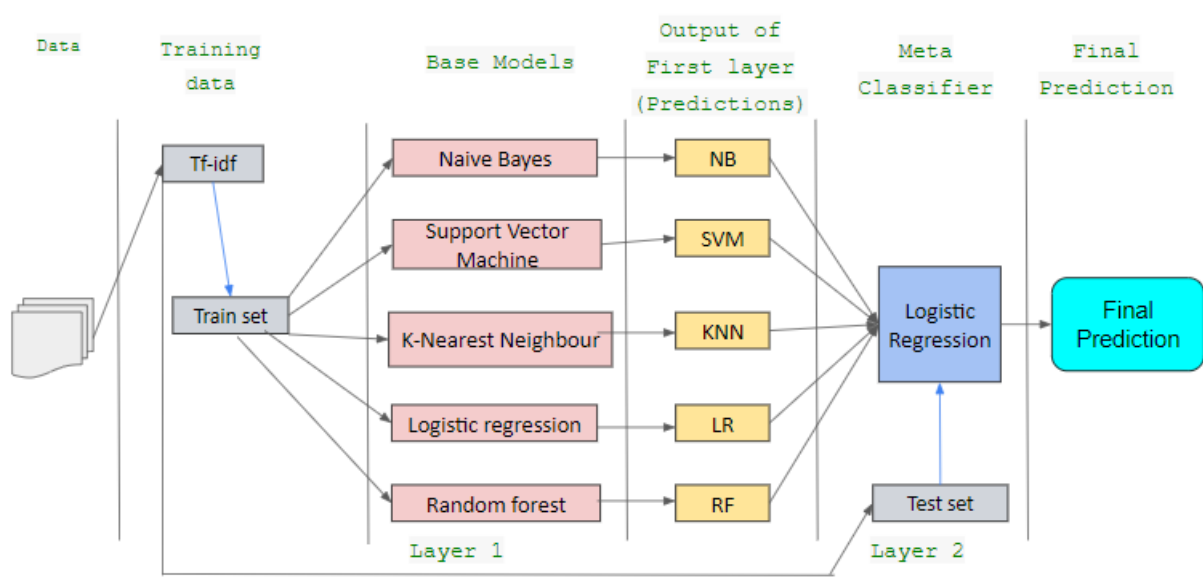


Figure 3.12: Work process of Stacking Classifier model

In this study’s architecture, the Stacking Classifier approach boils down to the clever deployment of Logistic Regression as the last predictor. This is the most significant for Logistic Regression because Logistic Regression is good at integrating the different predictions from the base models and, with the use of the logistic function, eventually comes with a single-linear model for new data instances. The Logistic Regression model takes its place within the ensemble as more than an instrumental but rather the indispensable link where the individual predictions from the ensemble are considered and merged, representing the summation of the ensemble modeling approach.

3.6.8 Stacking Classifier With Feature Engineering

More feature engineering techniques are used to improve the Stacking Classifier scores such as word vectorization as well as TF-IDF. This makes the existing methodology better at giving accurate scores. This analysis is meant to shed light on the improvements that are seen in terms of the accuracy, precision, fi-score, and feature engineering methods that have been developed in previous work.

Initially, the research did not include TF-IDF for feature extraction but instead developed more primary techniques. Besides, the previous approach which is considered as the first step in text mining was based mostly on surface-level interpretations and was limited in its ability to target the available set of textual nuances.

The use of Word2Vec and TF-IDF together will contribute to the methodological part. Unlike word-to-word mapping, where only relationships between specific words are understood, the vector representation provided by Word2Vec contains semantic relationships, based contextual use of words and offers a richer, more subtle understanding of textual data. The use of TF-IDF (Term Frequency and Inverse Document Frequency), which weighs the importance of the terms by considering the frequency analysis, not only adds one more dimension for representing words by their semantic characteristics but also adds yet an important layer of relevance to the previously computed word vectors. Feature engineering was an important part of the preprocessing phase where a feature set was formed by the combination of the semantic depth of the Word2Vec and the TF-IDF and so on, which allowed for the creation of a powerful model and the effective analysis of the obtained data. The first ensemble model could not go beyond the simple representation of small

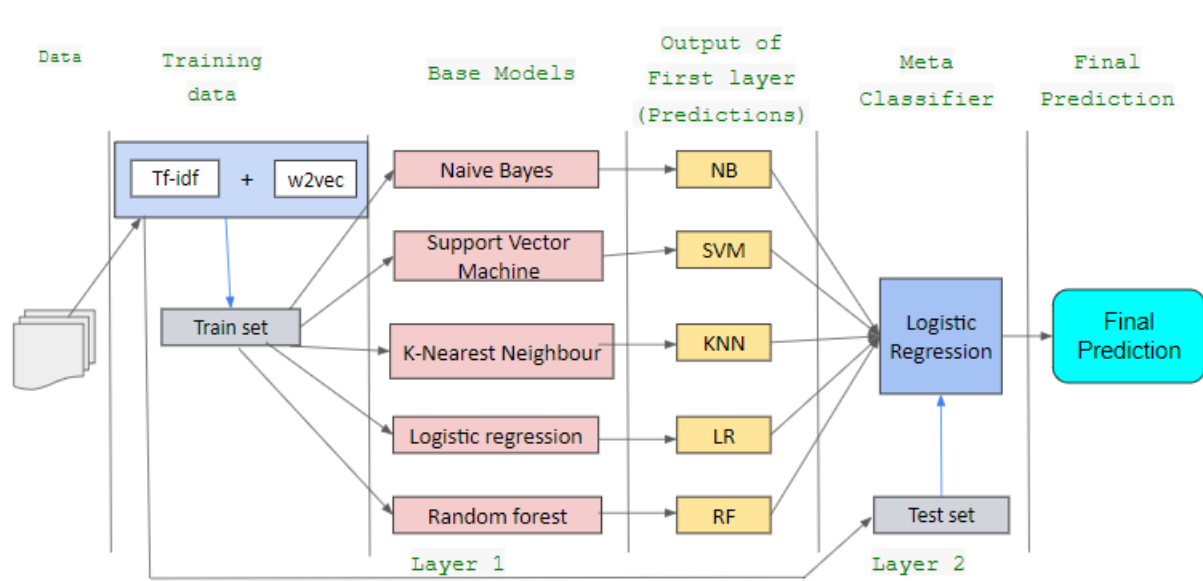


Figure 3.13: Work process of Stacking Classifier model

features in the absence of complex representation. The use of the more advanced feature engineering technique, which has included the use of Word2Vec and TF-IDF, has fundamentally changed the foundation of the ensemble model. The refinement of this idea was precisely implemented during model training, where the increased feature set inhabited a better environment for the Logistic Regression final estimator to “understand” the dataset. This dataset thus becomes richer and stronger, which constantly feeds the mechanism to classify, thus resulting in an enhanced accuracy rate and model capability. This is the purpose behind applying the strategy that stands out and justifies the significant difference in the model performance. Through the introduction of Word2Vec and TF-IDF feature selection algorithm, Stacking Classifier is now a feature set that is still at the same time whole and semantically rich, but with becoming more relevant. This methodology that is used has led to the development of a classifier that is stacking and which surpasses the previous one as it can give the text the desired accuracy. The growth in efficiency implies that this model is more suitable and complementary in the incorporation of

inconsistent outputs.

Feature extraction is the original approach which becomes advanced when word2vec and Tf-Idf have been used. Therefore, the study on Stacking Classifiers will be involved in a methodological evolution. Through the transition where an actual instance of feature engineering is performed as a part of the two phases preprocessing and model training, the threadbare approach is no longer time-consuming and the model has undergone a dramatic increase in the level of accuracy of the text classification tasks. The difference cases such as the process of adding more advanced feature engineering methods is an outstanding way to improve ensemble modeling capability in the sense of making precision and refinement techniques faster and more effective because of deep, contextual, and relevance-weighted features.

3.6.9 XGBoost Hybrid Model

In this model, at first, we split the pre-processed data into a train set and a test set. Then we use the train data for tf-idf vectorization and RoBERTa embeddings. TF-IDF (Term Frequency-Inverse Document Frequency) is a statistical measure that evaluates the significance of a phrase in a document relative to a corpus. After making use of TF-IDF vectorization to the documents, PCA is used to reduce the dimensionality of the TF-IDF matrix. PCA transforms the excessive-dimensional TF-IDF matrix right into a lower-dimensional space while preserving the most important information. On the other hand, RoBERTa is a pre-trained language model that generates contextualized word embeddings. In RoBERTa embedding, we tokenize words or subwords. Then we apply word embedding vectors and add them with positional embeddings. This vector is passed through feedforward neural networks, and finally pooling operations like mean pooling or max pooling are applied to obtain a single fixed-size representation.

Once the RoBERTa embeddings and the TF-IDF capabilities with PCA were acquired for each document, they were concatenated together. This concatenation effects combined feature vectors for every document, wherein the RoBERTa embeddings and the TF-IDF functions are stacked together. The concatenated characteristic vectors function as the input to the XGBoost classifier. XGBoost is a gradient-boosting algorithm that operates on tabular data. It takes the mixed function vectors as input and learns to predict the goal labels based on the patterns within the feature space.

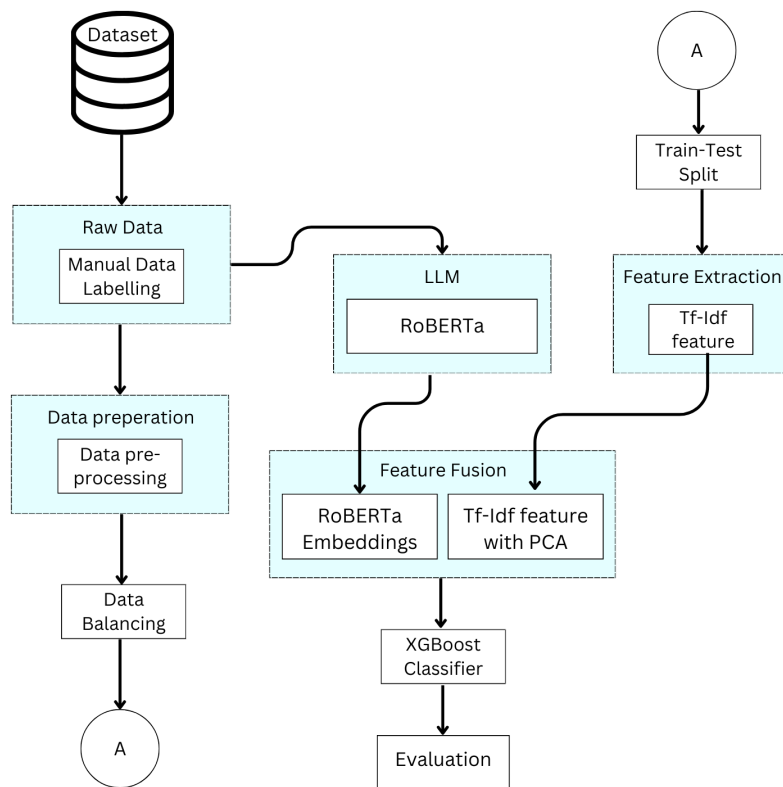


Figure 3.14: XGBoost hybrid model

Chapter 4

Result and Discussion

4.1 Training Performance

In this section, We will showcase and analyze the outcomes of our machine learning models designed to identify propaganda in poster titles. We present the test set’s accuracy, precision, recall, and F1 score for each model, and conduct a comparative analysis of their performance against both each other and the baseline model. In addition, we examine the confusion matrices and the feature importance scores of the models, allowing us to ascertain the strengths and weaknesses of each model. Subsequently, we analyze the findings in connection with our research issue and goals, and deliberate on the consequences, constraints, and suggestions arising from our investigation. In the development of the machine learning (ML) model, a robust validation process was employed to ensure its reliability and accuracy. We utilized a k-fold cross-validation technique, specifically a 5-fold validation, to assess the model’s performance. This method involves dividing the dataset into five subsets, training the model on four subsets and validating on the remaining one in each iteration. The process was repeated five times, and the average performance metrics were computed. The application of k-fold cross-validation enhances the model’s robustness by providing a more comprehensive evaluation across different subsets of the data, minimizing the risk of overfitting.

Model Name	Accuracy Score	Precision	Recall	F1-score
Random Forest	85.89%	84.00%	72.00%	75.00%
Logistic Regression	82.58%	84.00%	62.00%	64.00%
KNN	82.40%	78.00%	64.00%	67.00%
Naive Bayes	84.32%	87.00%	65.00%	69.00%
SVM	85.37%	83.00%	71.00%	74.00%
Stacking Classifier-1	87.11%	84.00%	76.00%	79.00%
Stacking Classifier-2	90.24%	87.00%	81.00%	84.00%
RoBERTa	86.06%	88.00%	67.00%	71.00%
Xgboost Hybrid	87.98%	85.00%	71.00%	75.00%

Table 4.1: Accuracy table of Machine Learning Models

To select the Stacking Classifier model, which is integrated with feature engineering techniques including the TF-IDF and Word2Vec, as the top-performer of our classification tasks, we compared this model against a number of other models men-

tioned below: RoBERTa, Random Forest, Logistic Regression, SVM, and others. The reason for opting for the Stacking Classifier model is its powerful arsenal, which include its capability to codify textual data by examining the features contributing to semantics along with the ones contributing to importance aspects. Due to this multi-pronged method, a more comprehensive notion of the text is acquired and in turn more achievement is achieved across various the dataset. Our classifier ensemble

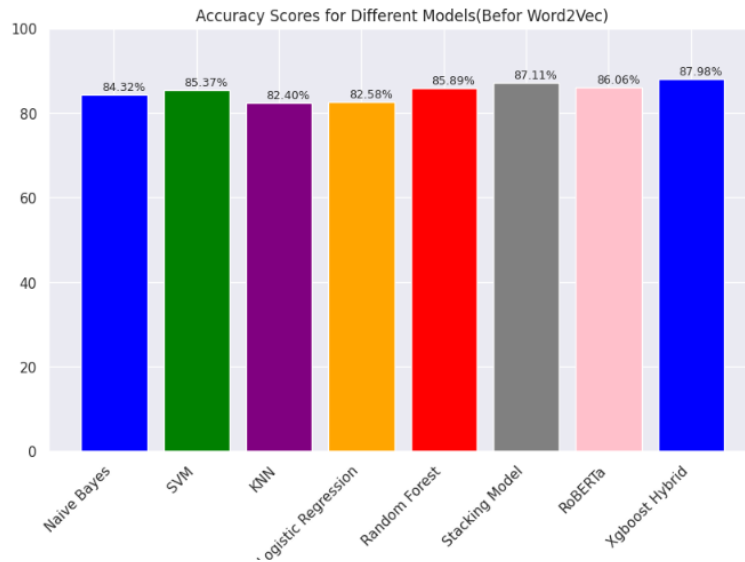


Figure 4.1: Classifier performance comparison before feature engineering

bles using the Stacking Classifier as the base model comes out to be the best option because we conducted a comparative analysis of the other available models. The following results compared the Stacking Classifier to the other models, and shown to have the best performance measured by the accuracy and the ability to deal with new data. This superiority is attributed to several key factors:

Advanced Feature Engineering

The combination of Word2Vec and TF-IDF characteristics gives way to a multidimensional representation of text data, taking into account not only their frequencies but also having them contextually in their sense. This approach by using the subject-object sentence vectorization and word embeddings not only helps to acquire a deeper understanding of the text itself, but also directly influences the model's performance in terms of predictive capability.

Leveraging Ensemble Strengths

The Stacking Classifier in essence exploits the specialization of individual base models in different ways, and leverages on pooled intelligence of these models by putting them together. This combined approach circumvents the disadvantages of each model while utilizing their advantages, hence, producing a resilient and adaptable model.

Strategic Meta-Classification

Letting Logistic Regression be the core classifier in the stacked learning algorithm provides a very good integration of base model outputs. This managerial decision taps into distinctions already known to Logistic Regression in the linear combination of all feature inputs so that the model can be steered about making the best output compensation as precisely as possible.

Customization and Flexibility

Although the single-model approach provides for a high level of customization and detail, the Stacking classifier with feature engineering offers much more flexibility and versatility. It provides an aggregate adaptation of both feature engineering processes and base models, which makes it versatile enough to address a wide range of text categorization tasks.

The drastic throughout analysis uncovers Stacking Classifier performance statistics and illustrates that this model excels in both of these areas. It stands for this as it is highly capable of working with multiple sophisticated feature representations alongside their predictive models, all of which make it the best among all the evaluated models that are suitable for dealing with complex text classification tasks.

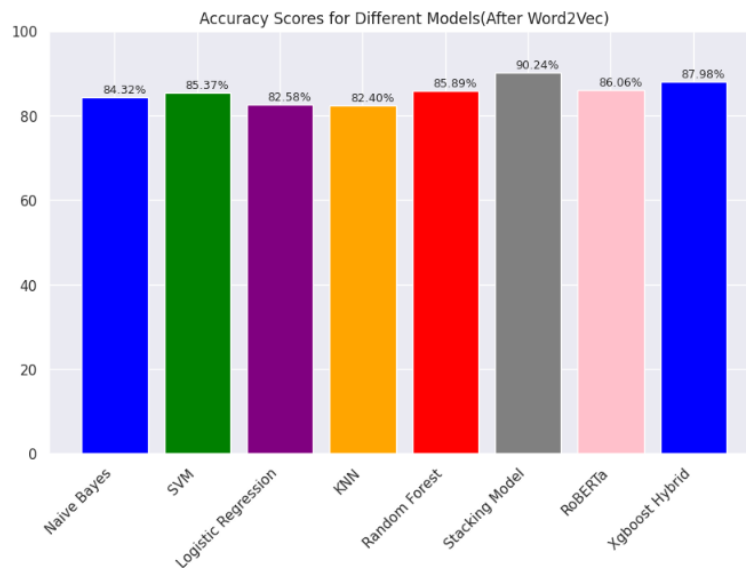


Figure 4.2: Classifier performance comparison after feature engineering

4.2 Machine Learning Models

So, initially there was 3056 data but after data pre-processing, our dataset is 2870 in total. So 20% of 2870 is $= (2870 * 20) / 100 = 574$ From 574 data, Naive Bayes has identified 444 non-propaganda and 40 propaganda with success. 4 data were projected as propaganda even though they were non-propaganda, and 86 data were forecasted as non-propaganda even though they were propaganda. SVM has successfully identified 434 non-propaganda and 56 propaganda. However it detected 14 data as propaganda even though they were non-propaganda and 70 as non-propaganda even though they

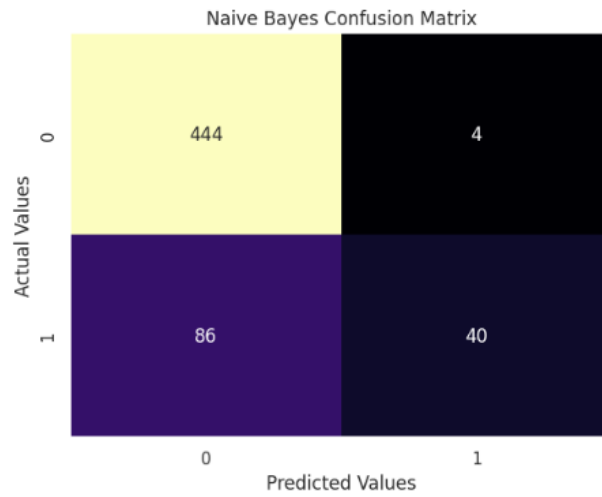


Figure 4.3: Naive Bayes Confusion Matrix

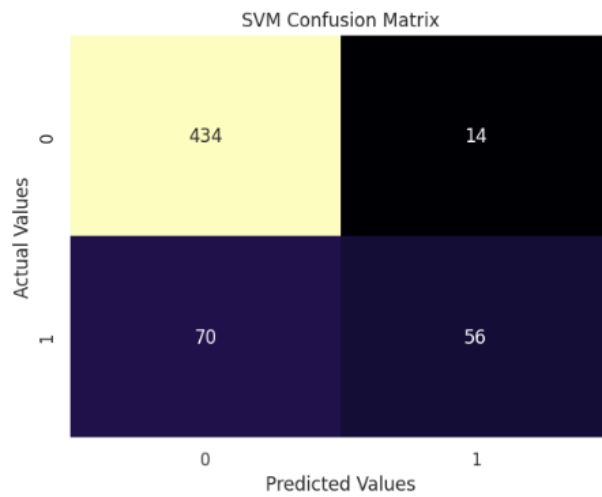


Figure 4.4: SVM Confusion Matrix

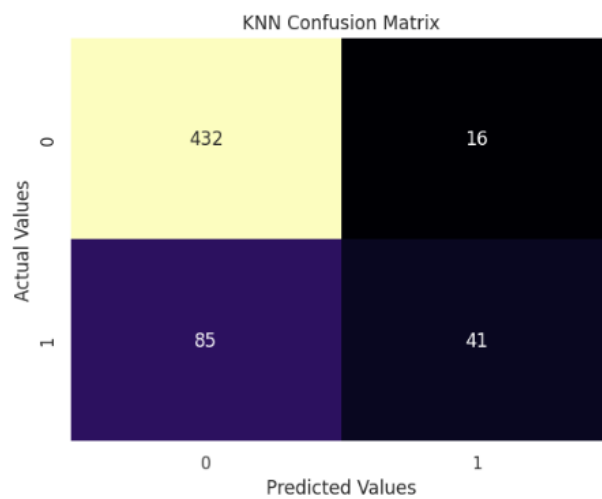


Figure 4.5: KNN Confusion Matrix

were propaganda. On the other hand, KNN identified 432 non-propaganda and 41 propaganda with success, whereas 16 data detected as propaganda even though they

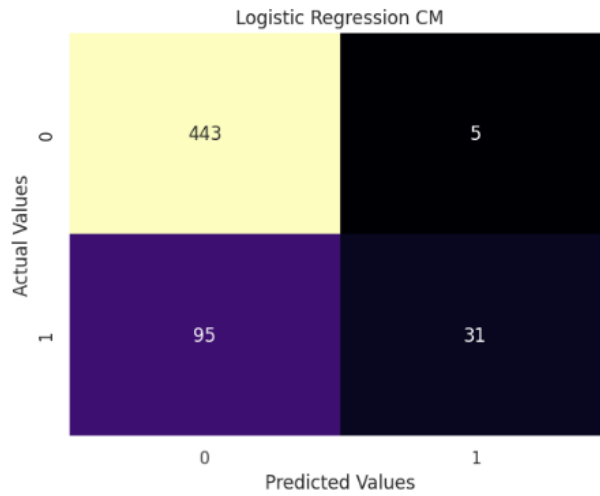


Figure 4.6: Logistic Regression Confusion Matrix

were non-propaganda and 85 as non-propaganda even though they were propaganda.

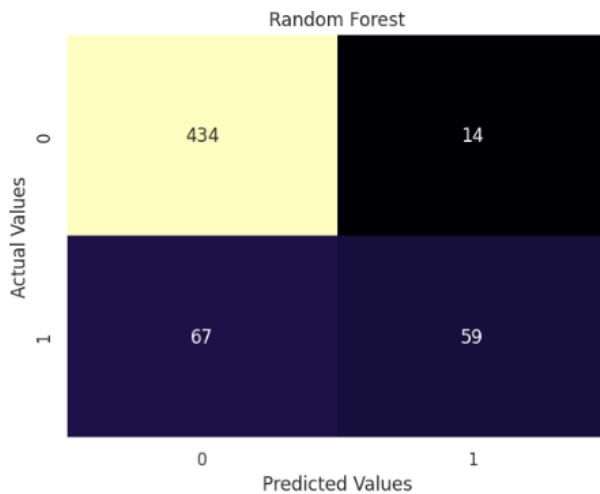


Figure 4.7: Random Forest Confusion Matrix

Logistic Regression detected 443 data as non-propaganda and 31 as propaganda. Impressively it detected only 5 data as propaganda even though it was non-propaganda and 95 as non-propaganda even though they were propaganda.

Random Forest has identified 434 non-propaganda and 59 propaganda with success. 14 data were projected as propaganda even though they were non-propaganda, and 67 data were forecasted as non-propaganda even though they were propaganda.

Here in Figure 4.7, Stacking Classifier is the ensemble model and we have implied feature engineering combined with Word2vec and Tf-Idf. It has successfully identified 441 non-propaganda and 77 propaganda. It also detected 16 data as propaganda even though they were non-propaganda and 40 as non-propaganda even though they were propaganda. But it detected more accurately than other models. Again in the Figure 4.8, We have done the previous process without feature engineering and it has successfully identified 429 non-propaganda and 71 propaganda. It also

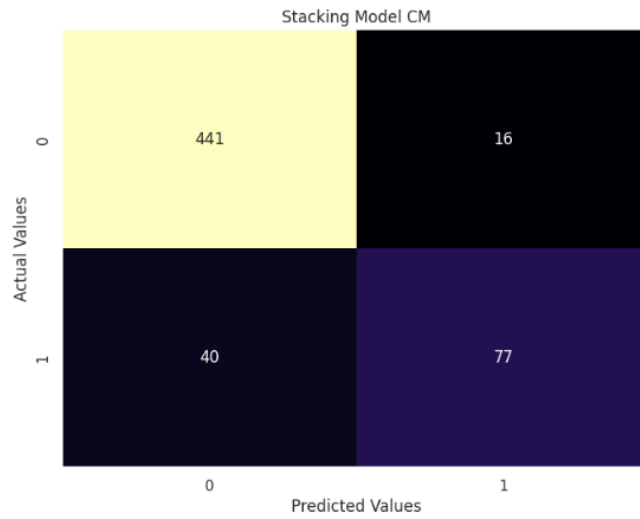


Figure 4.8: Stacking Classifier Confusion Matrix (Word2Vec + Tf-idf)

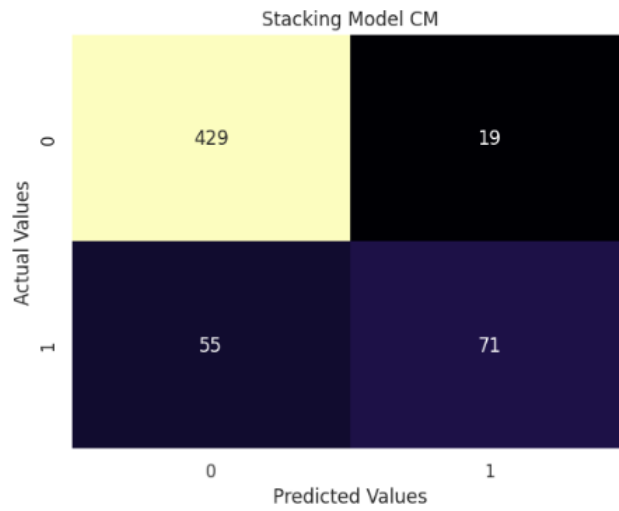


Figure 4.9: Stacking Classifier Confusion Matrix

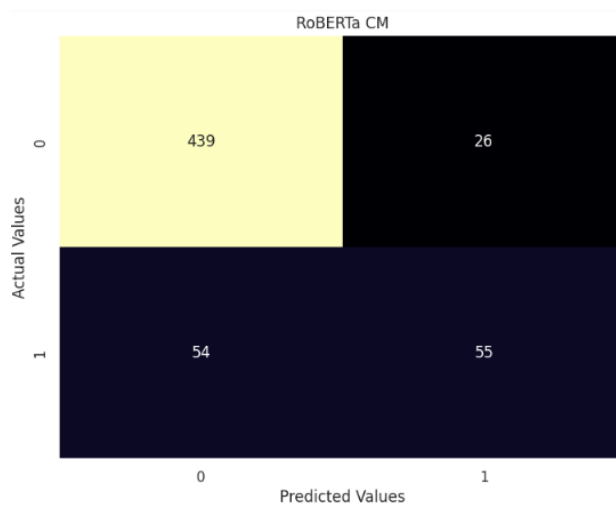


Figure 4.10: RoBERTa Confusion Matrix

detected 19 data as propaganda even though they were non-propaganda and 55 as non-propaganda even though they were propaganda. Comparing the variance, it shows better results when we use feature engineering.

Model Name	True-Negative	True-Positive	False-Negative	False-Positive
Naive Bayes	444	40	86	4
Logistic Regression	443	31	95	5
KNN	432	41	85	16
Random Forest	434	59	67	14
SVM	434	56	70	14
Stacking Classifier (W2vec+TF-IDF)	441	77	40	16
Stacking Classifier	429	71	55	19
RoBERTa	439	55	54	26

Table 4.2: Confusion matrix of Machine Learning Models

Lastly, RoBERTa has identified 439 non-propaganda and 55 propaganda with success. However, it detected 26 data as propaganda even though they were non-propaganda and 54 as non-propaganda even though they were propaganda.

To verify if the Stacking Classifier with Tf-Idf and Word2Vec performs better than other models, we checked the precision, recall, and F1-score values, which are 87.00%, 81.00%, and 84.00%, respectively. These values are also better than those of other models.

Stacking Classifier with Tf-Idf and Word2Vec stands out as a technique which has proved to be superior, in particularly when we take attention to feature engineering aspects. Eventually with high accuracy, precision, recall, and F1-score this model becomes an efficient instrument for propaganda detection through title of the posters.

Chapter 5

Conclusion

5.1 Conclusion

While contributing to the research work, we went through different works that have been done so far in the propaganda detection field and surveyed the relevant methodologies in propaganda-related works that relate to our domain. Later, we worked with our data and implemented different ML models to come out with the expected result we wanted. Further, we argued about the best-fitted ML model for the propagandistic poster title identification process and justified our best call for the best ML technique in this whole system. Finally, concrete auspicious directions in the field of propagandistic poster titles were claimed by us for the betterment of our domain.

5.2 Future work

As we brought up the challenges in the field of propaganda detection in the previous section, our future plan regarding the research is to overcome those challenges during our progression and achieve as following:

1. To enrich the dataset with more poster titles that can be of any type and train our system with those.
2. To add different languages poster titles in the dataset which can ensure versatility of the system and modify the system according to.
3. Extracting poster title through image processing and extending the dataset automatically.
4. Finally, making the whole system automated can help the mass people to detect any kinda of propaganda that is implicitly hidden in posters.

Bibliography

- [1] Institute for Propaganda Analysis, “How to detect propaganda,” *Propaganda Analysis*, vol. 1, no. 2, pp. 5–8, 1937.
- [2] J. Ellul, *Propaganda: The History, Art and Technique of Persuasion*. Vintage Books, 1965.
- [3] A. Barron-Cedeño, I. Jaradat, G. Martino, and P. Nakov, “Proppy: Organizing the news based on their propagandistic content,” *Information Processing Management*, vol. 56, no. 05, 2019.
- [4] G. Da San Martino, S. Yu, A. Barron-Cedeño, R. A. Petrov, and P. Nakov, “Fine-grained analysis of propaganda in news article,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 5584–5594. DOI: 10.18653/v1/d19-1565. [Online]. Available: <https://doi.org/10.18653/v1/d19-1565>.
- [5] G. Da San Martino, S. Yu, A. Barrón-Cedeño, R. Petrov, and P. Nakov, “Fine-grained analysis of propaganda in news article,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds., Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 5636–5646. DOI: 10.18653/v1/D19-1565. [Online]. Available: <https://aclanthology.org/D19-1565>.
- [6] P. Gupta, K. Saxena, U. Yaseen, T. A. Runkler, and H. Schütze, “Neural architectures for fine-grained propaganda detection in news,” *arXiv (Cornell University)*, 2019. DOI: 10.48550/arxiv.1909.06162. [Online]. Available: <https://doi.org/10.48550/arxiv.1909.06162>.
- [7] Y. Liu, M. Ott, N. Goyal, *et al.*, *Roberta: A robustly optimized bert pretraining approach*, 2019. arXiv: 1907.11692 [cs.CL].
- [8] N. Mapes, A. White, R. Medury, and S. Dua, “Divisive language and propaganda detection using multi-head attention transformers with deep learning bert-based language models for binary classification,” in *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, Association for Computational Linguistics, Nov. 2019, pp. 103–106.

- [9] G. Vlad, M. Tanase, C. Onose, and D. Cercel, “Sentence-level propaganda detection in news articles with transfer learning and bert-bilstm-capsule model,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019. DOI: 10.18653/v1/d19-5022. [Online]. Available: <https://doi.org/10.18653/v1/d19-5022>.
- [10] G.-A. Vlad, M.-A. Tanase, C. Onose, and D.-C. Cercel, “Sentence-level propaganda detection in news articles with transfer learning and BERT-BiLSTM-capsule model,” in *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, A. Feldman, G. Da San Martino, A. Barrón-Cedeño, C. Brew, C. Leberknight, and P. Nakov, Eds., Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 148–154. DOI: 10.18653/v1/D19-5022. [Online]. Available: <https://aclanthology.org/D19-5022>.
- [11] S. Yoosuf and Y. Yang, “Fine-grained propaganda detection with fine-tuned bert,” in *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, Association for Computational Linguistics, Nov. 2019, pp. 87–91.
- [12] G. Da San Martino, A. Barron-Cedeño, H. Wachsmuth, R. A. Petrov, and P. Nakov, “Semeval-2020 task 11: Detection of propaganda techniques in news articles,” *arXiv (Cornell University)*, 2020. DOI: 10.48550/arxiv.2009.02696. [Online]. Available: <https://doi.org/10.48550/arxiv.2009.02696>.
- [13] Federal Bureau of Investigation. “FBI Releases the Internet Crime Complaint Center 2020 Internet Crime Report Including COVID-19 Scam Statistics.” (2020), [Online]. Available: <https://www.fbi.gov/news/press-releases/fbi-releases-the-internet-crime-complaint-center-2020-internet-crime-report-including-covid-19-scam-statistics>.
- [14] D. Jurkiewicz, Ł. Borchmann, I. Kosmala, and F. Gralinski, “Applicaai at semeval-2020 task 11: On roberta-crf, span cls and whether self-training helps them,” in *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, International Committee for Computational Linguistics, Dec. 2020, pp. 1415–1424.
- [15] G. D. S. Martino, S. Shaar, Y. Zhang, S. Yu, A. Barron-Cedeño, and P. Nakov, “Prta: A system to support the analysis of propaganda techniques in the news,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Association for Computational Linguistics, Jul. 2020, pp. 287–293. [Online]. Available: <https://www.aclweb.org/anthology/2020.acl-demos.35>.
- [16] S. Yu, G. Da San Martino, M. Mohtarami, J. Glass, and P. Nakov, “Interpretable propaganda detection in news articles,” *arXiv (Cornell University)*, 2021. DOI: 10.48550/arxiv.2108.12802. [Online]. Available: <https://doi.org/10.48550/arxiv.2108.12802>.