ABUSIVE LANGUAGE DETECTION ON SOCIAL MEDIA

RAJKUMAR RAJENDRAPRASAD PAL

FINAL THESIS

NOVEMBER 2022

ABSTRACT

Since internet access has reached to everyone in their hands through mobile phones we have moved into the tech era or internet era where internet has become the basic necessity of livelihood, with the internet social media has also grown humongous, so much so that majority of internet usage is the part of social media usage. Just as selfies, dog and cat pics and other good things abusive and offensive content has also made their way to social media.

Using machine learning algorithms we try to find the underlying pattern and separate abusive content from the ones which aren't. Although the creation of a general metadata architecture has received minimal attention from researchers, abusive text classification on a Twitter dataset has remained a hot topic of study. The automated detection of hate speech and associated phenomena is the subject of a sizable corpus of study. We will be using Davidson English tweet dataset for this study.

We aimed to differentiate the tweets which contain abusive language or hate speech from the one's which aren't by using Logistic Regression, Naive Bayes, Support Vector Machines, Random Forest and XGBoost so that the algorithms can be further used to clear the tweets from the content wall of the users to whom the abusive content should be hidden. We found out that Logistic Regression performed better than Naive Bayes, Random Forest and XGBoost with and without TF-IDF transformation, but Support Vector Machines outperformed every other ML models and the results we achieved were Accuracy Score: 91.45 %, Precision Score: 91.49 %, Recall Score: 91.45 %, F1 Score: 91.41 % and Accuracy Score: 90.36 %, Precision Score: 90.40 %, Recall Score: 90.36 %, F1 Score: 90.30 % were achieved with the TF-IDF transformed data.

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	**Definition**
ALBERT	A Lite Bidirectional Encoder Representation from Transformers
ALD	Abusive Language Detection
AraBERT	Arabic Bidirectional Encoder Representation from Transformers
ATT-CNN	Attention Convolutional Neural Network
AUC	Area Under Curve
BERT	Bidirectional Encoder Representation from Transformers
Bi-LSTM	Bidirectional Long Short-Term Memory
Blacklist	a collection of terms recognized to be harsh or offensive
BMW	Bayerische Motoren Werke AG
BOW	Bag of Words
BSF	Bayesian Scoring Function
CF	CrowdFlower
CNN	Convolutional Neural Network
CNNLSTM	Convolutional Neural Network Long Short-Term Memory
DHOT	Devanagari Hindi Offensive Tweets
DNN	Delay Neural Network
EDA	Exploratory Data Analysis
FCNN	Fuzzy-based Convolutional Neural Network

GA	Genetic Algorithm
GCN	Graph Convolutional Network
GloVe	Global Vectors
GRU	Gated Recurrent Unit
HASOC	Hate Speech and Offensive Content
IndicNLP	NLP for Indian Languages
KNN	K-Nearest Neighbour
LIWC2015	Linguistic Inquiry and Word Count 2015
LR	Logistic Regression
LSTM	Long Short-Term Memory
mBERT	Multilingual Bidirectional Encoder Representation from Transformers
ML	Machine Learning
MOH	Map Only Hindi
MOLD	Multilingual Offensive Language Detection
MOLD-DL	Multilingual Offensive Language Detection Using Deep Learning
MNB	Multinomial Naive Bayes
MTC	Multilingual Text Categorisation
MuRIL	Multilingual Representations for Indian Languages
NB	Naive Bayes
NLP	Natural Language Processing

NLTK	Natural Language Toolkit
OLID	Offensive Language Identification Dataset
PCA	Principal Component Analysis
POS	Part of Speech
RELU	Rectified Linear Unit
RF	Random Forest
RFC	Random Forest Classifier
RGCN	Relational Graph Convolutional Network
RLC	Rocchio Linear Classifier
RoBERTa	A Robustly Optimized BERT Pretraining Approach
RSGNN	Relation-Special Graph Neural Network
SAGE	Sparse Additive Generative Model
Sci-Kit	Scikit-learn
SCNN	Sequential Convolutional Neural Network
SOLID	Semi-supervised Offensive Language Identification Dataset
SOM	Self Organising Maps
SVM	Support Vector Machine
TF-IDF	Term Frequency-Inverse Document Frequency
TIF-DNN	Transform-Invariant Features Delay Neural Network
T-NER	Transformer-based Named Entity Recognition

XLM-RoBERTa	Cross-lingual Language RoBERTa

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 History:

The history of Abusive Language Detection goes way back to the rise of social media or even back to the rise of the email era. Many things we see now on social media were started with emails, but as emails are not a part of our study but social media is so, we'll focus on social media.

The expansion of the internet allowed for the emergence of online communication services such as CompuServe, America Online, and Prodigy, which exposed consumers to digital communication via email, bulletin board messages, and real-time online chatting. In 2001 Friendster surfaced online and drew millions of users and facilitated email address registration and basic internet networking, though this was only primitive to the ones we have now. In 2003, Myspace was launched and around 2006, it was the most visited website on the planet, spurred by users' ability to share new music directly on their profile pages. Weblogs, one more primitive form of digital social communication, gained popularity with the establishment of the LiveJournal publishing site in 1999. This corresponded with the introduction of the Blogger publishing platform by Pyra Labs, a software startup bought by Google in 2003. In 2006 our favourite microblogging website was founded by Jack Dorsey, Evan Williams, Biz Stone, and others as Twitter. By 2008 a new networking website was around the corner by Harward alumni, which was named Facebook, since then Facebook has grown and widened its reach to become a conglomerate by acquiring multiple other companies in and out of the same domain.

Whenever we get something new or when we enter into a new place or when we meet a new person at first we try to be well-behaved because that's what we have been taught by our parents, but once we get hold of something or when start knowing the things or person little better, our behaviour shift from being polite to casual and after some more time in it, some more days with the same person we start treating it or them for granted. Similarly with social media people played the same game, when social media was new in the block we tried our best to be well-behaved and polite with the person we had chatted with, with comments we did on someone’s post or with the group we were part of. Soon after sometime when we got hold of social media we shifted our focus from being polite to being casual and over time it became granted and that’s how we can track the use of Abusive Language on Social Media. So in the early phase of social media, these websites rarely or didn’t have abuse or abusive language detection because it wasn’t needed back then but by the time the people changed but the websites didn’t in the same fashion so there came the problem with the rise of usage of abusive language on social media.

1.2 Background of the Study:

In this modern era as time progresses, we’re moving into the digital era. Now the internet has become more important than food, people can live a day without food but not the internet. With an increase in usage of the internet crime, abusiveness, bullying and hate speech have also increased all over the internet (Ayo et al., 2020) and parents wouldn’t want their children to learn this from the internet which they will eventually learn, the truth be told but not before time.

Talking about the internet, the majority of the internet usage is over social media and bullying, abusiveness and hate speech has also found their way to social media, (Fortuna et al., 2021) people these days are trying to fast to reduce their intake of social media, but eventually if there’s abusiveness we will take it deep down on our subconscious mind, and if fed to young child imagine what future we have when these children become adult and have their own children. (Ayo et al., 2021).

Social media provides a forum for people to communicate their opinions, expertise, experiences, and feelings, but it becomes a serious issue when these interactions are used as a vehicle for abusive statements, comments, and dialogues. Racism, sexism, and other ideologies may all be promoted through the use of abusive language. Here in this research we will be working on a dataset with tweets, twitter is a microblogging platform that empowers users to communicate with others through their tweets which are of 140 characters, hence it produces large amounts of data from brief digital material.

So in this research work, I'm trying to find the abusive language or words on social media, which can be further used to prevent it from being fed to a young child or even to elderly people, in short, it will be a small part in a bigger picture to prevent abusiveness from social media and trying to make the clean social media experience (Biradar et al., 2021; Khan et al., 2021; Ali et al., 2022; Wadud et al., 2022). We want to help establish a culture of civility online by using artificial intelligence.

Machine Learning can assist us with taking care of issues that were troublesome or incomprehensible previously, and abusive language detection is one of them. We want social media to be a safer environment for everyone, so we're taking the first step toward that goal and Machine learning is our most effective weapon against it.

1.3 Problem Statement:

Text classification and abusive language detection are not something new, there are methods involving doing several monolingual operations on datasets in Chinese and English. Abusive language detection is a difficult task that requires a lot of machine learning and pattern recognition algorithms. The detection is not always accurate and sometimes, it can be wrong and result in false positives. There are many challenges such as detecting slang, sarcasm, screaming, long sentences and many others. In a different study, the author puts out a system for categorising works in which contents were translated into a common tongue. To categorise text, they employed the RLC, Naive Bayes, and KNN after using WordNet to link terms to concepts. The co-regularization and consensus-based self-training methods that the author combined to study MTC don't include translation but instead looked towards the WordNet associated with every language (Anand et al., 2022). Using N-gram techniques, the author tackled multilingual text classification. Spanish, Italian, and English were the three languages they studied at MTC. (El-Alami et al., 2022) began by making educated guesses about the study language before categorising it with Naive Bayes. A host of ML techniques, including SVM, Decision Trees, KNN, SOM, and GA, have more recently been utilised to address the MTC problem in both Hindi and English. To increase the accuracy of the procedure, respective authors employed several feature selection techniques. The effectiveness of a pattern-based method was compared to that of KNN, Random Forest, and SVM to identify sarcasm in tweets. The majority of these methods rely on pre-trained text classification models (Anand et al., 2022). The great bulk of research on this topic focuses on English, in part because English-language resources are more widely available. Recently studies were done on different languages including Greek, Arabic, Dutch, Danish, French, Portuguese, Italian, Turkish and Slovene which lead us to additional datasets and tools for the above languages which can be further used to narrow down research on a particular language of choice. In context with abusive language detection online, two questions were addressed viz., 1. Which is more important for cross-dataset generalisation, the models or the datasets? 2. Additionally, which model and dataset properties are crucial for generalisation at the final? (Fortuna and Nunes, 2019; Fortuna et al., 2021). After this research there were still a few works that remained, a potential area of research is the use of combined datasets with the use of a category conversion schema prior to the merging, allowing for more fine-grained categorization. Recent works have shown high performance on merged datasets (Silva et al., 2016; Schmidt and Wiegand, 2017; Charitidis et al., 2020; Jha et al., 2020; Fortuna et al., 2021; Khan et al., 2021; Ali et al., 2022; Arango et al., 2022; Sharma et al., 2022; Subramanian et al., 2022).

1.4 Aims and Objectives:

The foremost aim of this research or study is to propose and compare models that can do the job of finding abusive language or hate speech content in the given tweet with the help of machine learning models. The objectives of this research can be found by dividing the aim into further steps: The foremost part is to gather a dataset of English tweets, do pre-processing and to make it suitable for our ML models. Predict the statement if it has any abusive language or hate speech in it. Compare and evaluate the results of multiple models.

1.5 Research Questions:

An abusive language is a form of violence. It’s a problem that needs to be addressed. We have the ability to stop it, and Machine Learning can be our greatest tool in this study. How can we detect abusive language or hate speech on social media? The principal question we ask ourselves while doing our research is to find “Does the current tweet contain any abusive language or hate speech?”

1.6 Scope of the Study:

The following things are inside the scope of this study or research:

- To find the abusive language or hate speech in a tweet
- To suggest a better machine learning model to predict the tweet if it contains abusive language or hate speech based on its performance on the dataset.

The following things are not inside the scope of this study or research:

- To replace the abusive word with another word or * or #.
- To replace the sentence or tweet with something new with similar meaning.
- To suggest tweets without abusive words in them.

1.7 Significance of the Study:

The significance of this research is to contribute to social media to make it a safer and cleaner environment for kids and elderly people, so when they visit social media they would not be bombarded with tweets which contain abusive language that might upset the mood of the kid or elderly person or worst case teach them to use those abusive languages and spread it in the real world. With the help of machine learning, we’re able to automatically detect and filter out abusive language. Our goal is to create a safe environment for all our members because we believe in making the world a better place.

1.8 Structure of the Study:

In Chapter One, my main focus is to describe the scope of this study by covering an overview of our primary question regarding our research as well as the background of previous research in this field of study. The focus is also on the aims and objectives, the scope of the study, and the significance of the study.

In Chapter Two, I conducted a literature review on the topic of research to determine when and who conducted research on a relevant issue, as well as what their purpose was, as well as their strategy and technique in their work, and how it can affect my study.

Chapter Three consists of the methodologies that were used by me to address the aims and objectives of this study, and I also shed some spotlight on the dataset with how I did pre-processing on the dataset before feeding it to the machine learning algorithms for learning and prediction also I cover what were my evaluation techniques and how well the models performed.

CHAPTER 2

LITERATURE REVIEW

We're on a mission to make the world a better place, and we believe it starts with creating social media a safer place for everyone. To do this in the modern world we use Machine Learning algorithms to detect and stop the use of abusive language on social media.

2.1 Need for abusive language detection:

Since social media became a part of our lives and we started using it daily instead of once a week or fortnightly, we because casual about it and then people started to talk or comment just as they would do in their real lives and started to use abusive languages, bullying, harassment, scamming, etc., When one interacts in internet usage, whether on web forum discussions and comments, or social media websites, there is always a major danger of scorn and even harassment. (Djuric et al., 2015; Nobata et al., 2016). Although social media allows people to communicate their opinions, expertise, experiences, and feelings online, a huge issue arises when social media interactions become a forum for hateful remarks, comments, and dialogues. (Jha et al., 2020). Administering abusive material across all user-generated text on the internet is a difficult task. However, the majority of tweets on Twitter and posts, or comments, or messages on Facebook are non-offensive and informative. Nonetheless, few people may be offensive to users or individuals or minority groups. Specific items are designated as offensive text if they are intended to be defamatory, humiliating, or insulting. In contrast to assaulting text, abusive material is typically targeted toward social classes such as gender, religion, handicap, sexual orientation or towards ethnic origin. (Davidson et al., 2017; Wadud et al., 2022). To prevent the use of abusive languages, many social media companies have standards guidelines that every users must adhere to, as well as using human editors in coupling with technologies that leverage blacklists and regular expressions to detect and eliminate inappropriate posts. As individuals increasingly converse online, the demand for high-quality automatic abusive language classifiers grows rapidly (Nobata et al., 2016; Silva et al., 2016).

Popular instances demonstrate the negative impact of harsh words in online forums and on social media. Facebook was chastised in 2013, for hosting groups or pages that were anti-women, such as “*Violently raping your friend just for laughs*” and “*Kicking your girlfriend in the fanny because she won’t make you a sandwich*”. In days, a campaign including over 200,000 signatories was launched, and numerous large corporations either withdrew or threatened to withdraw their advertising from Facebook after they were unintentionally placed on these pages or groups. Facebook isn’t the only corporation dealing with these issues; every company that contains user-generated material will have moderating concerns. This demonstrates the significant influence that hateful comments can have on a community as well as a major corporation. When actor Robin Williams deceased, his daughter Zelda wrote a tribute to her late father and was instantly abused on Twitter and Instagram eventually, she was forced to shut down all of her social media accounts. Because of this harassment, Twitter reviewed and revised its hate speech policies (Nobata et al., 2016).

2.2 Components of abusive language/Ambiguity with abusive language detection:

Abusive language detection is far more difficult than anticipated for a number of reasons. The noise in the data, along with the requirement for understanding global languages and their slang, makes this not just a tough task for automation, but also possibly a difficult assignment for individuals.(Nobata et al., 2016)

More than just targeted keywords. The deliberate entanglement of words and phrases in order to avoid machine or manual testing frequently makes detection difficult. Fabrications like “*ni99er*”, “*whoopiuglyniggerratgolberg*”, and “*JOOZ*” render standard keyword identification measures ineffective, especially when a source term or expression has multiple variants. Keyword detection, on the other hand, may result in false positives.(Nobata et al., 2016)

It is difficult to keep track of all racial and minority slurs. A blacklist (a collection of terms recognised to be harsh or offensive) can be used to create a somewhat good abuse or profanity classifier; however, these lists are not static and are constantly evolving. To keep up with linguistic changes, a blacklist would need to be updated on a regular basis. Furthermore, certain insults that are prohibited to one group but it might be perfectly acceptable to some group, thus the background context of the blacklist term is crucial.(Nobata et al., 2016)

Abusive language can be quite grammatical and fluent. Well there are a numerous examples of abusive language being rather loud on the internet, such as "*Add anotherJEW fined a bi$$ion for stealing like a tiny grub.*", "*Hang thm all*", which could be a useful signal for an automated system, there are numerous instances where abusive language, or particularly hate speech, is highly grammatical and fluent. For instance, "*I'm amazed they reported on this nonsense; who cares about another dead nigger?*"(Nobata et al., 2016)

Abusive behaviour can extend beyond sentence boundaries. In the phrase "*Chuck Hagel will protect Americans from the arguing of desert animals. Let them slaughter each other, good riddance!*". The success of the second sentence, which has the greatest repulsive intensity (slaughter each other), is dependent on their effective resolution to desert animals, which takes global knowledge to resolve. The lesson here is that harsh language does not stop with the phrase or a single sentence. In some circumstances, the other phrases must be considered in order to determine if the text is abusive or contains instances of hate speech.(Nobata et al., 2016)

Another ambiguity in abusive language identification is sarcasm. There are instances where some individuals would submit humorous remarks in the same tone as others who were using abusive language. This situation is extremely difficult for people or machine learning algorithms to accomplish right since it necessitates knowledge about the community and maybe even the users themselves: "*since I am disabled and must stay at home. I despise Jews because they ran over my legs with their BMW, so I'm going to bomb them every day with my posts.*", "*I'm cripple but I can destroy them with my posts.*", "*I'm a chicken, so I can post behind Yahoo's anonymous poster wall.*", "*I'm going to give him ten thumbs down and insult Jews. Bwbwbwbahahahah...I'm Hitler reborn.*" (Nobata et al., 2016).

Because of the interconnections between these components, researchers have grouped them under the umbrella labels of “abusive language,” “harmful speech,” and “hate speech,” but little research has been conducted to investigate their connections. Because each of these subtasks aims to address a distinct but somewhat overlapping dilemma, we feel there is much to be gained by investigating how they are interconnected. The diversity of labels used in previous work demonstrates the overlap across subtasks. For example, (Nobata et al., 2016) classify comparable remarks as “hate speech” or “derogatory language.” (“Waseem and Hovy, 2016”) evaluate simply “hate speech” without respect for any possible crossover with bullying or even other abusive languages, while (Davidson et al., 2017) separated hate speech from all other categories of offensive languages.

Unexpressed abusive language is defined as language that doesn’t instantly convey or signal abuse. The genuine nature is frequently concealed in this context by using ambiguous phrases, sarcasm, a lack of profanity or hostile language, and other ways, making it even more difficult to identify by both annotators and machine learning algorithms. Implicit, and even unconscious, acts of abuse known as “micro-aggressions”, have lately received more attention from social scientists and activists.

2.3 Related Works:

2.3.1 Related works in English language:

“The separation of hate speech from other instances of objectionable language is of significant difficulty for automatic hate-speech detection on social media. Lexical detection approaches have poor accuracy since they identify all communications containing certain phrases as hate speech, and earlier work employing supervised learning failed to differentiate between the two classifications” (Davidson et al., 2017). They gathered tweets containing hate speech terms using a crowd-sourced hate speech lexicon. They employed crowdsourcing to classify a sample of these tweets into three categories: those having “hate speech”, those containing “just offensive language”, and those featuring neither. “They trained multi-class classifiers to differentiate between these various groupings. A closer inspection of the prediction models and mistakes reveals when they can consistently distinguish hate speech from other offensive words and when this distinction is more difficult. They discovered that racist and homophobic tweets are more likely to be labelled as hate speech, but sexist tweets are more likely to be classified as offensive. Tweets that lack clear hateful words are likewise more difficult to categorise" (Davidson et al., 2017).

(Nobata et al., 2016) The majority of previous research on the subject of abusive language detection has really been dispersed among multiple overlapping domains. This might be confusing since various works may address different parts of abusive language, interpret the word differently, or apply it just to certain online domains (Twitter, online forums, etc.). To make comparisons across techniques even more difficult, practically all past work has used separate scoring sets. One of the accomplishments of this study is the provision of a public dataset to help move the field further. Since most basic systems rely on predetermined blacklists, it should be emphasised that some blacklist terms may not be abusive in the correct context. They demonstrated an increase in profanity detection using lists and an edit distance measure in their study. The latter enabled them to capture non-standard phrases like "*@ss*" or "*sh1t*". Another involvement of the effort was the use of crowdsourced to highlight offensive language. They employed Amazon Mechanical Turk workers to classify 6,500 online comments as abusive or not abusive. They only included comments when the majority agreed on the classification and only 9% of the comments contained abusive language. They also leveraged crowdsourcing efforts to compile a collection from several thousand online comments. The significant distinction is they did not limit the duty to simply abusive language but additionally have workers annotate for other sorts of hate speech and abusive language. To supplement other pertinent traits, their presentation is learnt using solely unigrams. By integrating several low-profile qualities, they intended for a system that is economical and versatile while still operating at high precision.

For online social media, automatic abusive language identification is a challenging though critical job. Their study contrasts a two-step technique of conducting classification on abusive language and then classifying it into particular categories with a one-step procedure of completing a single multi-class classification for identifying sexist and racist languages. With a public English Twitter corpus of 20 thousand tweets including sexism and racism, our technique achieves a promising F-measure of 0.827 using HybridCNN in one step and 0.824 using Logistic Regression in two phases (Park and Fung, 2017).

As the number of studies on abusive language detection and evaluation expands, there is an increasing need for critical examination of the links between the many subcomponents that have been classified under the term abusive language. Based on the research on hate speech, cyberbullying, and online abuse, (Waseem et al., 2017) suggested a typology that reflects the key similarities and distinctions across subcomponents and addresses the consequences of data annotation and characteristic generation. They stress the practical steps that scholars may take to a better approach in their abusive language detection subtask.

The results and key conclusions of SemEval-2019 Task 6 on recognizing and categorising Offensive Language in Social Networks are presented (OffensEval) by (Zampieri et al., 2019b). The project was based on the "Offensive Language Identification Dataset (OLID)", a new dataset containing around 14,000 English tweets. It has three subcomponents. The purpose of subcomponent A was to distinguish between offensive and non-offensive postings. The focus of subcomponent B was on the type of offensive content in the online posts. Finally, under subcomponent C, systems had to identify the intended recipient of the offending posts (Zampieri et al., 2019b).

Social media users may post anything they want without any supervision or limitation over the material, which results in a rise in the propagation of nasty and insulting speech among users, culminating in a surge in crimes, murder, and terrorism. As a result, (Alrehili, 2019) presented a comprehensive and state-of-the-art Natural Language Processing (NLP) approach for automatically identifying hate speech on social media such as dictionaries, bag-of-words, N-gram, and so on. People's material might range from good content with great value for others to hateful language. This distinction is due to the flexibility and absence of constraints that prohibit abusive language. Many online social media companies are trying to offset the impact of unrestricted material by incorporating algorithms that detect and severely ban abusive speech. Despite all efforts, identifying abusive language automatically remains a challenging task due to the multifaceted nature of the language (Schmidt and Wiegand, 2017; Alrehili, 2019; Fortuna and Nunes, 2019; Kaur et al., 2021). One of the most destructive utterances that have lately emerged on online social networking platforms is hate speech. Hate speech is defined as a dialect that targets a person or group based on a set of characteristics such as ethnicity, religion, gender, race, national origin, gender orientation, sexual orientation, and impairment. Merriam Webster defines hate speech as “a speech expressing dislike of a certain group of people.” Furthermore, hate speech is defined as “words meant to insult, offend, or intimidate a person due to a specific characteristic (national origin, sexual orientation, handicap, religion, or race).”

“The absence of a generalised framework, imprecision, threshold settings and fragmentation concerns are the main challenges for automatic hate-speech categorisation on Twitter. Most research employed binary classifiers to categorise hate speech, however, these classifiers are incapable of capturing other sentiments which may crossover between positive and negative classes. To address the challenge of hate speech categorization, a probabilistic grouping model for hate speech categorization in Twitter was created. A metadata extractor was used to gather tweets containing hate speech keywords, and crowd-sourced experts were used to categorise the hate tweets collected into two categories: hate speech and non-hate speech. The Term Frequency-Inverse Document Frequency (TF-IDF) model was used to represent features, which was supplemented with themes determined using a Bayes classifier. To automatically categorise real-time tweets into the appropriate classifications. Fuzzy logic was employed to classify hate speech leveraging semantic fuzzy rules and a score computation module. According to the assessment results, the constructed model significantly outperformed in hate speech identification, with an F1-score of 0.9256 utilising a 5-fold cross-validation” (Ayo et al., 2021).

Similarly, when compared to comparable models, the generated model for hate speech categorization made significant gains, with an F1-score of 91.5. When compared to previous methodologies, the proposed model also reveals a more perfect test with an AUC of 0.9645. The Paired Sample t-Test verified the created model’s effectiveness for hate speech categorization (Ayo et al., 2020, 2021).

“Because of the fast growth of online social media, Abusive Language Detection (ALD) has become a trendy topic in the area of affective computing. However, most ALD approaches in social networks ignore the interaction links between user postings, instead seeing ALD as a challenge of text context representation learning. To address this issue, we suggest a pipeline method that considers both the scope of a post and the features of the interaction network within which it is published. Our technique is split into two components: pre-training and downstream tasks. To begin, we employ Bidirectional Encoder Representation from Transformers (BERT) as an Encoder to construct phrase reconstructions in order to capture subtle contextual characteristics of the postings. Later, we construct a Relation-Special Network utilising the semantic relation of posts and the structural information from the interaction network. Upon that premise, they develop a Relation-Special Graph Neural Network (RSGNN) to effectively deliver information in the interaction network and learn text categorization. The experiment shows that their strategy can significantly enhance the identification of abusive posts across three public datasets. The results indicate that introducing interaction network structure into the abusive language detection task improves detection outcomes considerably” (Song et al., 2022).

“A substantial corpus of research has been conducted on the automated detection of hate speech and associated issues. Cross-dataset model generalisation, on the other hand, remains a difficulty. Throughout this setting, we discuss two still unanswered crucial questions: (i) to what degree does generalisation rely solely on the model and the composition and annotation of the training data in terms of different categories? And (ii) do particular features of datasets or models impact generalisation possibilities? To answer (i) they conducted intra- and cross-dataset experiments with BERT, ALBERT, fastText, and SVM models trained on nine standard public English datasets with standardized and comparable. The tests reveal that generalization differs between models and that some categories (for example, ‘toxic’ ‘abusive’, or ‘offensive’) work better as cross-dataset training categories than others (for example, ‘hate-speech’). To respond to (ii), they employed the Random Forest model to evaluate the importance of various model and dataset variables in predicting the performance of 450 BERT, 450 ALBERT, 450 fastText, and 348 SVM binary abusive language classifiers

(1698 in total). They discovered that in order to generalise effectively, a model must first perform well within an intra-dataset. Furthermore, several other characteristics, including the training and goal categories, as well as the fraction of out-of-domain vocabulary, are similarly important for the success of generalisation" (Fortuna et al., 2021).

(Zhang et al., 2018; Zhang and Luo, 2019)'s study describes a new strategy that is based on a deep neural network that combines convolutional and gated recurrent networks. They conducted a thorough assessment of the method against many of the benchmarks and state-of-the-art on the biggest collection of publicly released Twitter datasets to date, demonstrating that, especially in comparison to reported earlier results on these datasets, their suggested technique is capable of capturing both word sequence and order information in short texts.

2.3.2 Related works in Non-English language:

According to the number of speakers worldwide, Hindi is the third most spoken language in the world. It has been spoken by millions of Indians and has grown in complexity and usage as a result of varied geographical impacts and linguistic preferences. While "Hinglish" (Hindi written in the Roman script rather than the original Devanagari) is widely used online, the number of native Hindi speakers writing in Devanagari is increasing. Despite this, there have been few studies conducted on the usage of Hindi as an internet language. Their study proposes a methodology that uses a fastText based model to differentiate and then categorise abusive text from non-offensive text. The algorithm was able to categorise text from a dataset of Devanagari Hindi Offensive Tweets (DHOT). A grid-search strategy was used to modify hyper-parameters during fastText, which revealed intriguing insights into model accuracy and precision. Our fastText model obtained 92.2% accuracy when processed on a desktop-class computer. To the best of our knowledge, this is the first attempt to create a cutting-edge categorization of abusive language in Hindi utilising fastText models (Jha et al., 2020).

"When English is not the predominant language used on social media, individuals tend to post/comment in code-mixed text. This creates a number of difficulties in detecting offensive texts, and when paired with the limited available resources for languages such as Tamil, the work becomes even more difficult.” (Subramanian et al., 2022) Their study runs a series of experiments to detect potentially offensive text on YouTube comments, which are made accessible via the HASOC-Offensive Language Identification track in Dravidian Code-Mix FIRE 2021. “Models based on classic machine learning algorithms such as Bernoulli Naive Bayes, Support Vector Machine, Logistic Regression, and K-Nearest Neighbour were developed to detect offensive texts. Furthermore, pre-trained multilingual transformer-based natural language processing models like mBERT, MuRIL (Base and Large), and XLM-RoBERTa (Base and Large) were tried. These models were utilised as fine-tuners and adapter transformers. Adapters and fine-tuners achieved the same purpose, in essence, however, adapters work by adding layers to the main pre-trained model and freezing their weights. According to this research, transformer-based models surpass machine learning methods. Furthermore, adapter-based methods outperform fine-tuned models in terms of both time and efficiency in limited resource languages such as Tamil. Among all adapter-based techniques, XLM-RoBERTa (Large) had the greatest accuracy of 88.5%.”

(Sharma et al., 2022) proposed study focuses on analysing hate speech in a code-switched Hindi-English dialect. They investigated transformation strategies in order to obtain accurate text representation. They created “MoH”’ meaning (Map Only Hindi), which means “Love” in Hindi, to contain data structure while still allowing it to be used with existing methods. The “MoH” pipeline includes language recognition, Roman to Devanagari Hindi transliteration utilising a knowledge base of Roman Hindi terms, and lastly the fine-tuned Multilingual Bert and MuRIL language models. They performed many quantitative trial investigations on three datasets, measuring performance with Precision, Recall, and F1 measures. The first trial compares the performance of “MoH” mapped text with standard machine learning models and finds an average gain of 13% in F1 scores. The second measures up the proposed work’s scores against those of the baseline methods and finds a 6% performance improvement. Finally, utilising the current transliteration library, the third evaluates the suggested “MoH” approach to various data simulations. “MoH” surpasses the others by 15% in this case. Code-mixing is a prevalent occurrence in social media writing, especially in multilingual nations like India. Traditional deep learning approaches learned on monolingual data do not work well enough on code-mixed data, and training the new models is difficult owing to resource constraints. Transforming multilingual data to monolingual is a critical solution to this problem. In this paper, TIF-DNN, a Transformer-based Interpretation and Feature Extraction Model, is suggested for hate speech classification. In our proposed model, we employed the IndicNLP and English To Hindi libraries for transliteration and translation, respectively, and mBERT for feature extraction (Biradar et al., 2021).

“While hate speech is a worldwide problem, most current experiments on automatic hate speech detection are carried out in a single language. In this paper, they investigated hate speech identification in low-resource languages using zero-shot learning to transfer information from a resource-rich language, English. The results reveal that their joint-learning models outperform in the majority of languages. However, a straightforward technique that employs machine translation and a pre-trained English language model provides reliable results. In contrast, Multilingual BERT does not do well in cross-lingual hate speech identification. We also discovered that external knowledge from a multilingual abusive lexicon can boost the models’ performance, particularly in recognising the positive class.” (Pamungkas et al., 2021).

This researcher (Alsafari et al., 2020) seeks to develop efficient Arabic hate speech and offensive language detection mechanisms. They created a trustworthy Arabic textual dataset by crawling data collected from Twitter and used four reliable extraction algorithms based on four categories of hatred: race, religion, gender and nationality. Following that, they labelled the corpus using a three-hierarchical annotation approach, ensuring ground truth at each level by verifying inter annotation compatibility. They created multiple 2 – class, 3 – class, and 6 – class classification models using machine and deep learning algorithms, which were integrated with a range of techniques for feature extraction such as contextual word embeddings. Finally, they performed a rigorous experiment to evaluate the performance of the various trained models and to investigate misclassification mistakes. Compared to the previous results on hate speech and offensive language research conducted on the Arabic language, their performance findings are quite promising.

(El-Alami et al., 2022) “suggested research that would use transfer learning models and the fine-tuning phase to handle the Multilingual Offensive Language Detection (MOLD) challenge. They propose an efficient method based on Bidirectional Encoder Representations from Transformers (BERT), which has demonstrated tremendous promise in collecting semantics and contextual information inside texts. The suggested method is divided into three stages: (1) preprocessing, (2) text representation using BERT models, and (3) classification into two groups: offensive and non-offensive. To deal with multilingualism, they investigated several strategies such as joint-multilingual and translation-based ones. The first is to create a single categorization system for all languages, and the second is to translate all writings into a single global language and then categorise them. They ran numerous tests on a multilingual dataset derived from the Semi-supervised Offensive Language Identification Dataset (SOLID). The experimental results reveal that the translation-based technique combined with Arabic BERT (AraBERT) obtains an F1-score of 93% and an accuracy of over 91%.”

The Bengali language has around 13,000 grapheme variants including several diacritics, vowels, consonants and consonant conjuncts. These grapheme variations also comprise several combinations of character patterns. (Wadud et al., 2022) research aims on assessing human created materials on social media sites employing the LSTM-BOOST computerised deep learning algorithm. Text from several social media sites is retrieved, pre-processed, and categorised into various sorts of objectionable labels. Their paper presents an LSTM-ADA Boost ensembled architecture as a Bengali abusive language text recognition technique for detecting abusive content on social media networks. Ensembled Long Term Memory-Adaptive Boost is the name given to the suggested system (LSTM-BOOST). It was trained using Bengali datasets that included 20,000 memes, posts and comments from various Bengali websites, blogs and social media networks.

2.4 Research Methodologies:

(Davidson et al., 2017) “First tested logistic regression with L1 regularisation to minimise the dimensionality of the data, followed by testing a range of models previously utilised: logistic regression, Naive Bayes, decision trees, random forests, and linear SVMs. The models were then validated using 5-fold cross-validation, with 10% of the data set aside for assessment to minimise over-fitting. They discovered that the Logistic Regression and Linear SVM performed much better than other models after utilising a grid-search to iterate through the models and parameters. After deciding on logistic regression with L2 regularisation for the final model since it allows us to assess the estimated probability of class membership more easily and has done well in earlier works (Waseem and Hovy, 2016). After training the final model on the complete dataset and using it to predict the labels for every tweet, utilising a one-versus-rest framework in which each class is trained separately and the class label with the greatest predicted probability across all classifiers is assigned to each tweet."

(Nobata et al., 2016) used a supervised classification approach that employs NLP characteristics to assess various parts of the user comment. In particular, they utilise Vowpal Wabbit's regression model in its standard configuration with a bit rate of 28 and based NLP features on previous work in sentiment analysis, text normalisation, and other areas. They divided features into 4 sub-components i.e. N-grams, Linguistic, Syntactic and Distributional Semantics. They applied some moderate pre-processing for the first three features to convert some of the noise in the data, which might affect meagre features in the model. Normalising numbers, replacing very lengthy unfamiliar phrases with the same token, replacing repetitive punctuation with the same token, and so on are examples of transformations. None of the aforementioned normalisations were conducted for the fourth feature class.

(Park and Fung, 2017) proposed three models of CNN based to identify sexist and racist abusive languages i.e. HybridCNN, CharCNN and WordCNN. The most significant distinction between these models relates to whether the input characteristics are characters, words, or both. The convolutional layers, each of which calculates a one-dimensional convolution over the previous input with numerous filter sizes and collective feature map sizes, are the crucial aspects. Having different filter sizes is equivalent to viewing a sentence via many screens at the same time. After the convolution, max pooling was used to collect the feature or features that were most important to the outcome.

(Alrehili, 2019) “Used hate speech methods widely on NLP methodologies. The building of dictionaries for the identification of hate speech was done in various methods and in various languages. Bag of words, like dictionaries, is another NLP approach used to identify the hateful and abusive language. After accumulating a series of words in the bag of words, word frequencies were chosen as a feature for training a classifier. The N-gram approach is the most often utilised in the automated identification of hate speech. N-gram is used not only to gather words, and it might be further used to capture characters or syllables of a word. This technique is not vulnerable to spelling alterations when it is used in words. The TF-IDF is a tool that determines the relevance of a word in a document inside a corpus. Term frequency signifies the term is significant when it occurs in numerous documents, but the opposite document frequency means a word appearing in many documents has less relevance. As a result, TF-IDF has assigned a high weight or relevance to a term that appears often in the text but is uncommon in other documents. The involvement of the word in context is revealed by the part of speech, which increases the significance of the context. The Facebook post is manually annotated by five unique annotators based on the given categorization. They used POS, sentiment polarity, word embedding lexicons, and morpho-syntactic characteristics, and they built and designed two classifiers. Moreover, they rely on employing two machine learning algorithms i.e. Long Short-Term Memory (LSTM) and Support Vector Machines (SVM). Rule-based techniques are extensively utilised in the field of text mining, which comprises a collection of rules developed by humans and augmented by linguistic expertise. The benefit of rule-based techniques is that they produce great accuracy when compared to other methods; nevertheless, manipulating systems based on rule-based approaches require a lot of time and effort. Sentiment analysis can also be referred to as opinion analysis or emotion analysis. In AI it is the systematic quantification, extraction, identification, and study of subjective information and affective states using computational linguistics, text analysis, natural language processing, and biometrics. The purpose of sentiment analysis is to determine a writer’s, subject’s, or speaker’s attitude toward an emotional reaction to an event, encounter, or document, or the overall contextual polarity of some subject.”

(Zampieri et al., 2019b) in their paper they have mentioned according to their sub-tasks and how many people used which method to perform and how good that algorithm performed altogether some also used ensemble methods. Sub-task A was perhaps the most popular, with 104 teams participating. Seven of the top ten teams employed BERT, with changes in the settings and pre-processing processes. "*NULI*", the top-performing team, employed BERT base-uncased with default parameters but with a maximum sentence length of 64 and trained for two epochs. "*MIDAS*", was rated sixth who used the top nonBERT model. They employed a CNN and BLSTM+BGRU ensemble, as well as Twitter word2vec embeddings and token/hashtag normalisation. A total of 76 teams took part in sub-task B, with 71 of them also taking part in sub-task A. Unlike in sub-task A, when BERT obviously won, five of the top ten teams utilised an ensemble model. "*Amobee*" and "*HHU*", the second and third teams, employed ensembles of deep learning (including BERT) and non-neural machine learning techniques.

(Fortuna et al., 2021) begin by developing binary intra-dataset classification models for BERT, ALBERT, fastText, and SVM, on nine datasets and predefined categories. They employed off-the-shelf models for BERT, ALBERT, and fastText, and they randomly divided the training sets of all datasets, except Offenseval and Hateval, which retained their original training and test sets, into 70% for training and 30% for testing. Instead of a 70%/30% split, a 10-fold cross-validation was employed for the SVM research findings. Stopwords were deleted for the Bag-Of-Words (BOW) extraction, and only words with a frequency higher than or equal to 1% were evaluated. Most of the default parameters for SVM classification were utilised, except for the kernel, which is set to the linear kernel. They employed SVM with bagging due to the time complexity of the parameter extraction and training operations.

(Wadud et al., 2022) suggested the LSTM-BOOST model made use of a modified AdaBoost method that included principal component analysis (PCA) and LSTM networks. The dataset was separated into three categories in the LSTM-Boost model, and PCA and LSTM networks were applied to each section of the dataset to collect the most significant variance and minimise the weighted error of the model's weak assumption. Furthermore, for the baseline experiment, several classifiers were utilised, and the model was tested using various word embedding vector approaches.

(Song et al., 2022) “Created a pipeline architecture that incorporates the context, semantics, and interaction network information in posts and converts the text classification problem into a node classification task by generating a relation-special graph. Regardless of the efficiency of network communications, learning the context elements of a tweet is the most straightforward approach to assessing whether it is abusive. Due to their remarkable capacity for self-attention and the collection of vast amounts of data, several pre-trained language models, such as BERT and RoBERTa, have obtained high outcomes in text representation learning. They employed HateBERT, a re-trained BERT model using a large-scale dataset of Reddit comments from communities banned for being unpleasant, abusive, or hateful, to better match the ALD field. It is natural to regard postings as nodes and design networks to inject the interaction network structure. As a result, they built a relation-special network with several sorts of edges G = { N, A1, A2 }. Because each user’s nickname is unique, they also constructed interaction connections by matching the string *@username*. If the same user name occurs in two separate tweets, they presumed that both tweets have a strong association and added an undirected edge between them. To calculate semantic similarity between various postings, they used word-based Term Frequency-Inverse Document Frequency (TF-IDF), one of the most used static text vector computing methods. Unlike regular GCN, they prioritised self-loop contributions in their RGCN-inspired method. HateBERT thoroughly examined the post-node characteristics as contextual features, thus they include finer semantic information that can aid in node categorization.”

(Ayo et al., 2021) “Divided their research process into four phases: metadata representation, training, clustering, and classification. The metadata representation includes a generic metadata extractor API for collecting social media data from many sources. For data pre-processing, the filtering technique, combinatorial algorithm, BSF, and semantic analysis are used during the training stage. The filtering technique is used to isolate noise from incoming tweets. Furthermore, the data pre-processing module’s objective is to differentiate hate speech from non-hate speech tweets using a combinatorial algorithm based on a normalised threshold value utilising BSF. The combinatorial approach was chosen since it is suited for text similarity and broad online detection systems. During the training phase, semantic analysis is also used to extract tweet characteristics that will subsequently be used as input in the classification module. Data representation and detection modules are used in the clustering step. The data representation module is used to generate topic models automatically using a probabilistic topic spotting metric based on the Bayes theorem. As training tales, the subject model is developed utilising a database of hate speech from the training phase. To avoid the problem of fragmentation associated with most internet clustering, the hate speech database was utilised as training data for topic grouping. The detection module is a rule-based system that uses a modified Jaccard similarity metric to cluster real-time tweets into any of the generated topic categories. The rule-based detection method assists in transitioning between iterative and incremental clustering for real-time hate tweet identification utilising a score normalisation method. The goal of score normalisation is to provide a single threshold value for each tweet-topic similarity. Finally, the classification phase involves a hate speech classification module based on characteristics retrieved during the training phase, and a user interface for system interaction. For a 4-level sentiment categorization of hate speech, the classification module employs fuzzy logic."

(Subramanian et al., 2022) "Used traditional machine learning approaches, such as Bernoulli Naive Bayes, Support Vector Machine, Logistic Regression, and K-Nearest Neighbour, were used to generate models. Furthermore, pre-trained multi-linguistic transformer-based natural language processing models like mBERT, MuRIL (Base and Large), and XLM-RoBERTa (Base and Large) were explored. These transformers were utilised as fine-tuners and adapters. Adapters and fine-tuners achieve the same purpose in essence however adapters work by adding layers to the main pre-trained model and locking their weights."

(Sharma et al., 2022) compared the initial performance to that of traditional machine learning models that utilised simple surface characteristics. The models were then evaluated on raw data and subsequently on suggested MoH mapped data. Second, the suggested work was compared to baseline models. Third, different data transliteration strategies were compared to the suggested technique. These data transliterations were learned using the same Bert-based models as were used to train MoH mapped datasets.

(Pamungkas et al., 2021) “Employed Logistic Regression in conjunction with LASER Embedding. This method performed well in cross-lingual hate speech detection, particularly for low-resource languages with small training datasets. They made use of the Scikit-Learn library’s default hyper-parameters. They used a cutting-edge model for numerous natural language processing tasks in English, namely the Transformer-based architecture BERT (bert-base-cased), which is accessible on TensorFlow-hub and allows BERT to be integrated with the Keras functional layer. Based on the Multilingual BERT neural model. This model also employs a pre-trained Multilingual BERT model from TensorFlow-hub (bert-multi-cased). The rest network design is identical to that of the BERT model, in that they stacked dense with RELU activation and dense with sigmoid activation. The adoption of the multilingual BERT model enables the architecture to receive the text in any language without the need for translation. This model is also tuned using Adam with a learning rate of 2 – 5. To adjust this model, batch sizes (32, 64, 128) and epochs (1 – 5) were used.”

(Alsafari et al., 2020) “Used the specified feature selection techniques to train three algorithms for each classification task: Multinomial Naive Bayes (NB), Support Vector Machine (SVM), and Logistic Regression (LR). And trained 15 multi-class classifiers (2-class, 3-class, and 6-class) on distinct feature spaces. Grid search and 10-fold cross-validation were used to modify the kernel type and regularisation parameter C for SVM, the alpha value for NB, and the regularisation and penalty parameters for LR. Each experiment was conducted 10 times using a randomised initialization and a segregated training and validation split. In particular, the training data is separated into training and validation sets (in an 80:20 ratio) with an equal balance of the classes for each iteration. They assessed the performance of the learnt algorithms on the testing datasets for each classification assignment.”

(Anand et al., 2022) research sought to address MOLD-DL (Multilingual Offensive Language Detection Using Deep Learning) methodologies as well as natural language processing in feature selection and classification. Fuzzy-based convolutional neural networks were used to pick features for segmented data (FCNN). The selected features were then extracted and classified using an ensemble architecture of the Bi-LSTM model with Naive Bayes architecture hybrid with Support Vector Machines (SVM).

(Khan et al., 2021) work is divided into two parts: an offline training module and online hate and offensive speech detection module. Offline training is a recurring task that collects tweets and identifies tweets tagged by various persons. The offline training technique instructs the deep neural network on how to recognise characteristics in tweets. The online approach is in charge of predicting the labels for new tweets by using the model developed in the offline approach. Users of social media can either agree or disagree with the automated labels. Tweets classified in the online method, as well as fresh tweets labelled by Twitter users, are put back into the offline procedure to re-train the algorithm for automated labelling job optimization. The suggested model for hate categorization was a sequential convolutional neural network (SCNN). SCNN is a sequential model featuring embedding, three convolutional 1-D layers and three max pooling 1-D layers, as well as Dropout, Flatten, and dense layers. Several models were evaluated using the SCNN technique for tweet categorization for hate speech. The following methods were evaluated: (i) I n-grams using SVM, (ii) LR with a list of multiple features, (iii) LSTM, (iv) CNNLSTM, (v) CNN-nonstatic, (vi) CNN2D, (vii) ATTCNN, and (viii) ATTCNN with max. For a valid comparison, n-grams up to n=4 in the first two methods and filter sizes ranging from 2 to 4 in neural networks were used. N-grams with SVM and LR with various feature lists were used as baseline models.

(Ombui et al., 2019) adopted a mixed research technique. First, the qualitative technique was employed through content analysis to determine the discriminant features of the hate speech phenomenon by examining important hate theories in literature as simply stated in the framework. The framework was subsequently turned into a web application, which nine human annotators used to categorise 48k short text messages (tweets) into one of three predetermined classes: Hate Speech, Offensive, or Neither. A quantitative method of text analysis was employed to get word frequencies per class, and then additional low-level characteristics such as TF-IDF and word frequency vectors were utilised to train the classifier model. To manage all operations, including data preparation, data visualisation, model training, and actual classification, a unified method was used. This was utilised to assist in end-to-end model creation and data visualisation.

(Zampieri et al., 2019a) picked SVM because it was the easiest machine learning model; it is a linear SVM trained on word unigrams, and it has previously produced state-of-the-art results for them on numerous text categorization problems. They also tried out a bidirectional Long Short-Term Memory (Bi-LSTM) model that we developed from an existing model for sentiment analysis. Finally, a Convolutional Neural Network (CNN) model was utilised, with the same multi-channel inputs as the preceding Bi-LSTM.

(Lee et al., 2018) “Built his work on unsupervised learning of abusive text with word embedding through word2vec’s skip-gram and cosine similarity between two-word vectors. To detect purposely obfuscated terms, the system additionally incorporates various effective gadgets to filter abusive material based on blacklists, n-grams, edit-distance, mixed languages, abbreviations, punctuation, and words with special characters. When the findings of the unsupervised learning module and the detection gadgets are identical, the word is added to the abusive or non-abusive word lists, this process is known as auto-tagging. When the unsupervised learning and the detection gadgets disagreed, human specialists were requested to assess the term based on the abusive word judgments standards and then insert the word in the associated list, this process known as manual tagging.”

(ElSherief et al., 2018) “Used numerous metrics based on previous researches to investigate linguistic elements that distinguish between Directed and Generalised hate speech. To mitigate the consequences of domain shift in model selection, they employed tools built and trained to utilise Twitter data when accessible and fall back to state-of-the-art models trained on English data if Twitter-specific tools are unavailable. SAGE, a mixed-effect topic model that implements the L1 regularised version of sparse additive generative models of text, was used to assess the salient terms for each category of hate speech keywords (e.g., ethnicity, class, gender) and particular linguistic semantics associated with hashtags. SAGE has been utilised in various Natural Language Processing (NLP) applications that try to analyse how attitudes shift temporally around the issue of slavery-related United States property law judgements. T-NER, a system created expressly to conduct Named Entity Recognition on tweets, is used to extract entities from the gathered tweets. They employed the psycholinguistic lexicon programme LIWC2015, a text analysis tool that analyses psychological dimensions such as attachment and cognition, to study the linguistic dimension and psychological processes identified among Directed hatred, Generalised hate, and generic Twitter tweets. SEMAFOR, which annotates text with its evoked frames as described by FRAMENET, was used to evaluate the frame semantics underlying hate speech."

(Gambäck and Sikdar, 2017) examined four hate-speech categorization techniques based on distinct feature embeddings. The English Twitter hate speech dataset was used to test all models. Each tweet in the dataset has been labelled with four labels: non-hate speech (84% of the data), racism, sexism, and both. This was done by one Expert annotator and three Amateur annotators. The system's initial phase is to produce feature embeddings. Word embeddings and character n-grams were used to create feature embeddings for all words.
The word embeddings were created using two methods: word2vec and random vectors. Along with the word embeddings, the one-hot character of length 28 n-gram vectors with 26 elements for the English alphabet, one for numbers, and one for all other characters/symbols were created. In the network, a pooling layer was employed to transform each Twitter into a fixed-length vector, collecting the information from the full tweet. The most relevant latent semantic features from the tweets are subsequently captured using a max pooling layer. A softmax layer produces the class probability distribution function on every tweet and allocates the hate-speech classes/labels depending on the probability values on the output side.

(Badjatiya et al., 2017) tested his hypothesis using a dataset of 16K annotated tweets. Of the 16K tweets, 3383 were sexist, 1972 were racist, and the remainder was neither sexist nor racist. They employed GloVe pre-trained word embeddings for the embedding-based algorithms. GloVe embeddings were learned using a huge Twitter dataset. They also tried using different word embedding sizes. In addition, 10-fold cross-validation was done, and weighted macro accuracy, recall, and F1-scores were produced. As optimizers, Adam was applied to CNN and LSTM, and RMS-Prop for FastText. CNN and LSTM training was done in batches of 128 for CNN and 64 for FastText.

2.5 Summary:

In recent years, there has been a spike in interest in detecting abusive language, hate speech, cyberbullying, and trolling. Social media platforms are also subjected to growing pressure to address these concerns. Because of the similarities between these subtasks, researchers have grouped them under the banner of “harmful speech”, “abusive language”, and “hate speech.” In this section of my study I have found that in recent past years this researchers who did their research in “abusive language detection” or “hate speech detection” some had used only English dataset while some has used multiple language dataset as we know this issue occurs not only in English language but other languages too, people used multiple machine learning algorithms to come to a conclusion and get a result out of their works.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction:

This section of this research paper is where we discuss about the dataset that is about to be used for the research work and the pre-processing methods that will be applied to the data before serving it to the machine learning algorithms to train and then predict. This section will also discuss the validation techniques or the parameters that will be used to test the performance of our models.

3.2 Research Methodology:

3.2.1 Dataset Description:

The dataset which we are using here is the Twitter English tweets dataset, also known as the Davidson dataset, it is found in the Repository for the paper "Automated Hate Speech Detection and the Problem of Offensive Language", ICWSM 2017. It is an English dataset with all the tweets in the English language. The dataset contains 24783 tweets, each tweet is either of hate speech or abusive language or neither as voted by the users of CrowdFlower.

The dataset contains 5 columns and they are:

1. count: "amount of individuals that coded each tweet on CrowdFlower (min is 3)."
2. hate_speech: "amount of users on CF who voted the tweet to be hate speech."
3. offensive_language: "amount of users on CF who voted the tweet to be offensive."
4. neither: "amount of users on CF who voted the tweet to be neither non – offensive nor offensive."
5. class: "class label for majority of users on CF. 0 – hate speech 1 – offensive language 2 – neither."

3.2.2 Data Pre-processing:

Data cleaning or pre-processing is essential as the original dataset is usually noisy and poorly customize on top of it, it might also have data that might be irrelevant to our objective. The dataset contains lots of data that are of no use to us, i.e. our objective is to find the abusive language or hate speech in the tweets.

3.2.2.1 Missing Value Treatment:

Handling missing values is the very first step in data pre-processing as there might missing values than might make our model inefficient and in many cases might result into error. The next step would be deduplication, this step would be helpful to identify the duplicate tweets that might influence our ML model to be custom.

3.2.2.2 Feature Selection:

After handling missing values, we will remove the features that are not associated with offensive or abusive language detection or any feature that will not be useful in model training.

3.2.2.3 Text Cleaning:

Once we have removed the features that are not required the next step would be to convert all the tweets into small cases and after conversion removes unnecessary words that are irrelevant to our research i.e. removal of punctuations, user handles or mentions, hashtags (#) and emoticons.

3.2.2.4 Tokenization and Normalization:

After this step, we need to tokenize and customize our tweets so that it becomes feed able to our models and so that models can identify the underlying patterns in the tweets.

3.2.2.5 Handling Imbalance Data:

Once we are done with all tokenization we also need to makes sure that the dataset is not imbalanced and we need to do resampling i.e. oversampling or undersampling so that the model can learn about the each class properly without overfitting anyone class.

3.2.3 Modelling:

In this study, our approach for the modelling will start with data gathering. Once we have the data the next step will be to clean it with pre-processing methods and analyse the data. Once the data is clean we can move to the next step by feeding our data to machine learning algorithms. Our approach for prediction will be to start from the bottom and move upwards i.e. to start from the basic model and then move on to complex models.

3.2.3.1 Logistic Regression:

With the help of a linear combination of one or more independent variables, the log-odds for the event are used in the statistical model of Logistic Regression to represent the likelihood that an event will occur. The mathematical formula for Logistic Regression is:

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

Figure 3.1: Logistic Regression formula

The Maximum Likelihood Estimation customize the Cross entropy loss function in machine learning applications when logistic regression is employed for binary classification.

3.2.3.2 Naive Bayes:

Naive Bayes classification algorithms are a type of basic “probabilistic classifier” that uses Bayes’ theorem with significant (naive) assumptions of independence across features. Naive Bayes classifiers are extremely scalable, with parameters that are proportional to the amount of features in a learning task. Instead of costly iterative approximation, maximum likelihood training may be performed simply evaluating a closed-form expression in linear time. The conditional probability may be deconstructed using Bayes’ theorem as

$$p(C_k \mid \mathbf{x}) = \frac{p(C_k)\, p(\mathbf{x} \mid C_k)}{p(\mathbf{x})}$$

Figure 3.2: Conditional probability using Bayes Theorem

Logistic regression classifiers supplant Naive Bayes classifiers on binary features.

3.2.3.3 Support Vector Machine:

Support Vector Machines are supervised learning models that use learning methods to examine information for classification and regression. SVMs, which are built on statistical learning frameworks or the VC theory introduced by Vapnik and Chervonenkis, are among the most resilient prediction methods. The hinge loss function is useful for extending SVM to circumstances when the data are not linearly differentiable.

3.2.3.4 Random Forest:

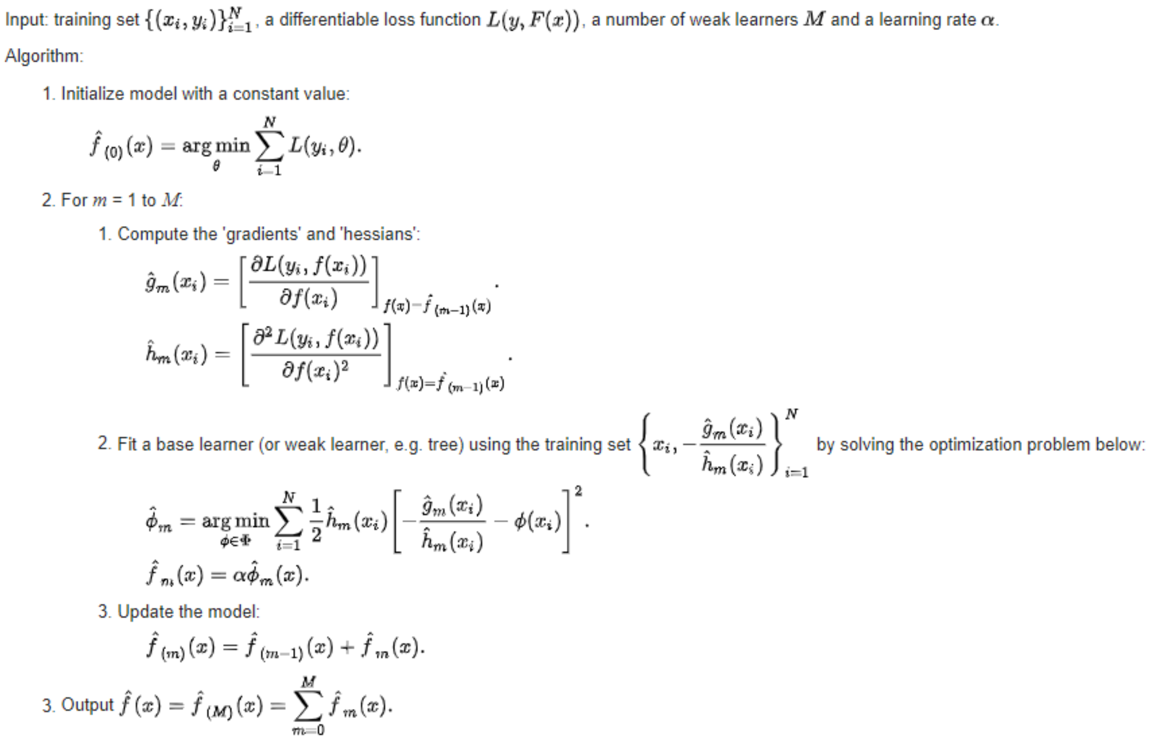
Random forests is an ensemble learning algorithm for classification, regression, and other problems that works by building a large number of decision trees during learning. For classification problems, the random forest outputs the class chosen by the majority of trees. After training, predictions for unseen data x’ may be produced in Bagging modelling techniques by summing the predictions from all the separate regression trees on x’ as

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b(x')$$

Figure 3.3: Prediction formula for bagging model

3.2.3.5 XGBoost:

Tianqi Chen created XGBoost as a research project as part of the Distributed (Deep) Machine Learning Community (DMLC). It started off as a terminal programme that could be customized with a libsvm configuration file. It gained popularity in the ML competition community after being used in the winning answer of the Higgs Machine Learning Challenge. While the XGBoost model frequently outperforms a single decision tree in terms of accuracy, it compromises decision trees' inherent readability. An unregularized XGBoost algorithm in general is,



```
Input: training set $\{(x_i, y_i)\}_{i=1}^{N}$, a differentiable loss function $L(y, F(x))$, a number of weak learners $M$ and a learning rate $\alpha$.
Algorithm:
  1. Initialize model with a constant value:
       $\hat{f}_{(0)}(x) = \arg\min_{\theta} \sum_{i=1}^{N} L(y_i, \theta).$
  2. For $m$ = 1 to $M$:
      1. Compute the 'gradients' and 'hessians':
           $\hat{g}_m(x_i) = \left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x)=\hat{f}_{(m-1)}(x)}.$
           $\hat{h}_m(x_i) = \left[\frac{\partial^2 L(y_i, f(x_i))}{\partial f(x_i)^2}\right]_{f(x)=\hat{f}_{(m-1)}(x)}.$
      2. Fit a base learner (or weak learner, e.g. tree) using the training set $\left\{x_i, -\frac{\hat{g}_m(x_i)}{\hat{h}_m(x_i)}\right\}_{i=1}^{N}$ by solving the optimization problem below:
           $\hat{\phi}_m = \arg\min_{\phi\in\Phi} \sum_{i=1}^{N} \frac{1}{2}\hat{h}_m(x_i)\left[-\frac{\hat{g}_m(x_i)}{\hat{h}_m(x_i)} - \phi(x_i)\right]^2.$
           $\hat{f}_m(x) = \alpha\hat{\phi}_m(x).$
      3. Update the model:
           $\hat{f}_{(m)}(x) = \hat{f}_{(m-1)}(x) + \hat{f}_m(x).$
  3. Output $\hat{f}(x) = \hat{f}_{(M)}(x) = \sum_{m=0}^{M} \hat{f}_m(x).$
```

Figure 3.4: XGBoost model in general

3.2.4 Evaluation:

We can see that selecting the appropriate model is also crucially important. Testing models on a dataset provides important information on the quality of the dataset in a nascent research field like abusive language detection, where so much subjectivity still remains. The evaluation metrics we will use in our study to evaluate the performance of our models are:

- Accuracy: "The probability of correctly classified abusive tweets from the total number of tweets present."
- Precision: "It is the probability of the predicted abusive tweets that were actually labelled as abusive."
- Recall: "It is the probability of abusive tweets correctly predicted as abusive tweets."
- F1-score: "The F1-score combines a classifier's precision and recall into a single statistic by calculating their symmetrical mean."

3.3 Proposed Model:

We will start classification of the tweets from Logistic Regression. Logistic Regression performs at a significant level for binary classification. Once we have some results from Logistic Regression, we can move on to SVM as it has been given state of the art result to many of the classification problems (Zampieri et al., 2019a). Once done with simpler models we move on to the ensemble methods and use Random Forest and see how they are performing on our train and test dataset. In our study the main objective is binary classification, so Naïve Bayes serves better with binary classification. Once that is done we can then finally move on to the complex XGBoost Classifier. In Random Forest and XGBoost, we will try hyper-parameter tunings that can give us different results and train our model to perform better on a test set.

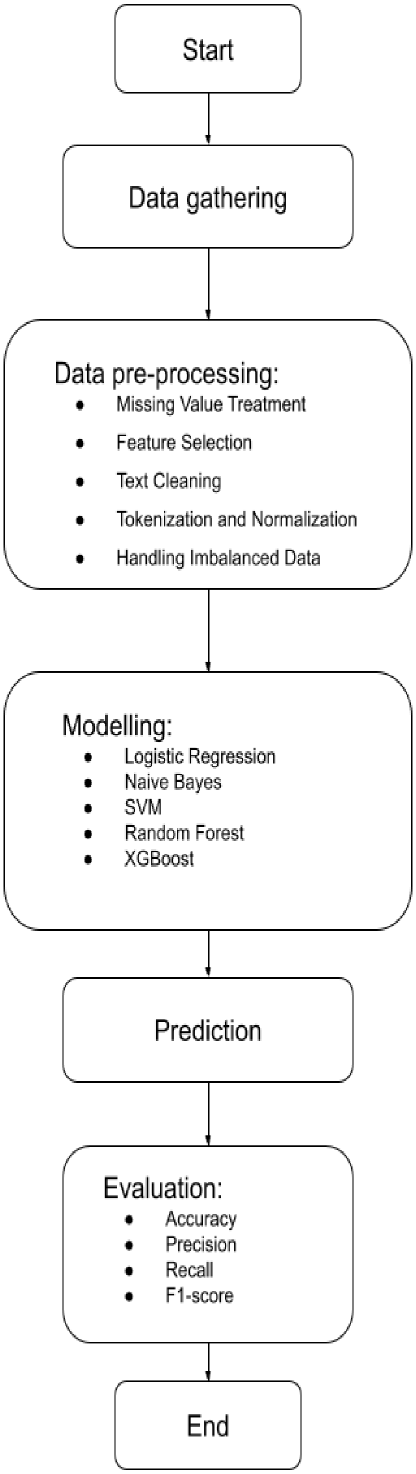


Figure 3.5: Modelling flow chart

3.4 Summary:

This section totally focuses on the methods we’ll be using to conduct the study. Here we first start with data gathering. Once we have data we also discuss the data pre-processing that is needed before we start serving our data to machine learning algorithms and then after we can feed our data to machine learning algorithms and start evaluating them.

CHAPTER 4

IMPLEMENTATION

4.1 Introduction:

This is the section where we discuss the actions, about what we did, how we did. Here we discuss about the dataset and the EDA we did to see the patterns in the data and after EDA what and possibly how we did our modelling and also which models we used to come to conclusion and end our study.

4.2 Dataset Description:

Understanding our dataset is the very first thing one should do when planning for a study on Machine Learning or Data Science. As our study is on the topic of Abusive Language Detection on Social Media so for that we have a Twitter English tweet dataset which is also known as the Davidson dataset, it can be found in the repository for the paper "Automated Hate Speech Detection and the Problem of Offensive Language", ICWSM 2017. This dataset contains 24783 tweets all the tweets are in English language, every tweet is either of hate speech or abusive language or neither as per the votes given by the users of CrowdFlower.

The dataset contains 5 columns:

1. count: "amount of individuals that coded each tweet on CrowdFlower (min is 3)."
2. hate_speech: ""amount of users on CF who voted the tweet to be hate speech."
3. offensive_language: "amount of users on CF who voted the tweet to be offensive."
4. neither: "amount of users on CF who voted the tweet to be neither non – offensive nor offensive."
5. class: "class label for majority of users on CF. 0 – hate speech 1 – offensive language 2 – neither."

4.3 Data Preprocessing:

4.3.1 Handling Missing Values:

The very first step we do in data preprocessing is to find which features and how many rows have the missing values. These missing values have a great impact on our model learning, so it is very critical to handle missing values. After checking isnull() we found out that the dataset didn’t have any null or missing value.

4.3.2 Feature Selection:

Dataset contained 6 features i.e. ‘count’, ‘hate_speech’, ‘offensive_language’, ‘neither’, ‘class’, ‘tweet’, now apart from ‘class’ and ‘tweet’ all other features are useless to our study as they don’t have any relation with the ‘class’, it’s only the ‘tweet’ we need to feed to our model to learn and be able to predict the ‘class’. So we will now drop ‘count’, ‘hate_speech’, ‘offensive_language’, ‘neither’ features from our dataframe to move ahead in our study.

4.3.3 Text Cleaning:

After the feature selection we get to the ‘tweet’ feature which consists of the tweets written by people and ‘class’ that is tagged as 0 for hate speech, 1 for offensive language and 2 for neither. Now before we need to feed tweets to our model we need to clean it and make it appropriate to be able to feed it to the models. So here we removed stop words, “RT”, http or https, numeric values, symbols, used porter stemmer and finally lowered the text case of all the words that were left. After cleaning the tweets now it can be used for next steps i.e. tokenizing or transforming to TF-IDF representations.

4.3.4 Tokenization:

Splitting the text into smaller elements is known as tokenization, so here we took the tweets and used TweetTokenizer() by sklearn to tokenize the tweets to be able to feed to our machine learning models. We used ngram_range from 1 to 5, max_features of 2000 and word as analyzer for hyperparameter tuning.

4.3.5 TF-IDF Representation:

To represent our words in a smarter way into matrix format we used a TF-IDF transformer. TF-IDF helps in assigning higher weights to terms that are present frequently in a document and which are rare in among all documents, on the other hand it assigns low score to terms which are common across all documents.

4.4 Exploratory Data Analysis:

Exploratory Data Analysis is the stage where we understand our before doing any preprocessing or modelling, we use statistics to understand the patterns that are already present in the dataset. Here we didn’t have many features, in total 6 features and out of which 4 features were dropped as they served no purpose to our study as they didn’t have any logical relation to our study with either ‘class’ or ‘tweet’. Now we are left with ‘class’ and ‘tweet’ and for our EDA we have only 1 feature namely ‘class’. We did some EDA on ‘class’ feature. First we created a count plot to see the bar of classes present into it, we found out that class 0 for hate speech had in total 1430 rows, class 1 for offensive language had 19190 rows and class 2 for neither had 4163 rows.

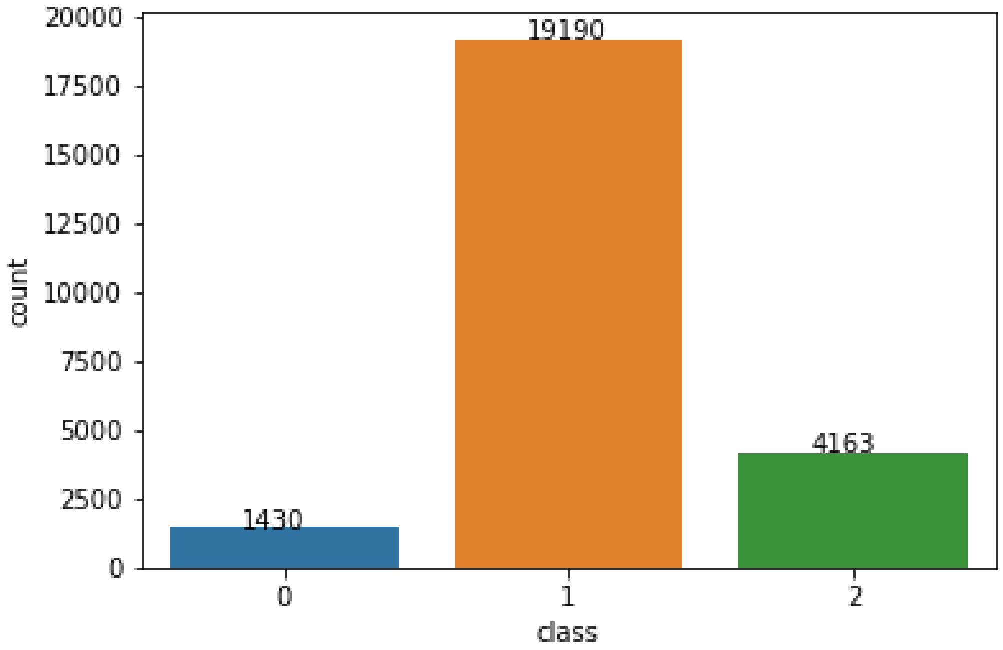


Figure 4.1: Countplot of class feature

Once we got the count of rows in each class after that, we separated hate_speech, offensive_language and neither tweets from each other and formed separate wordcloud of each to see the prominent words that were used in those classes.

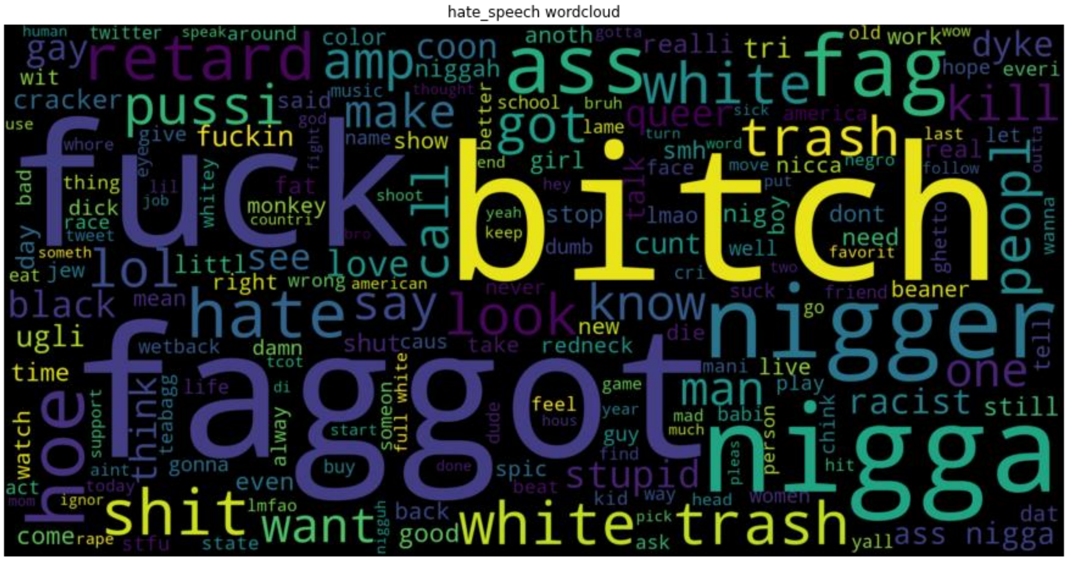


Figure 4.2: Wordcloud of hate_speech tweets

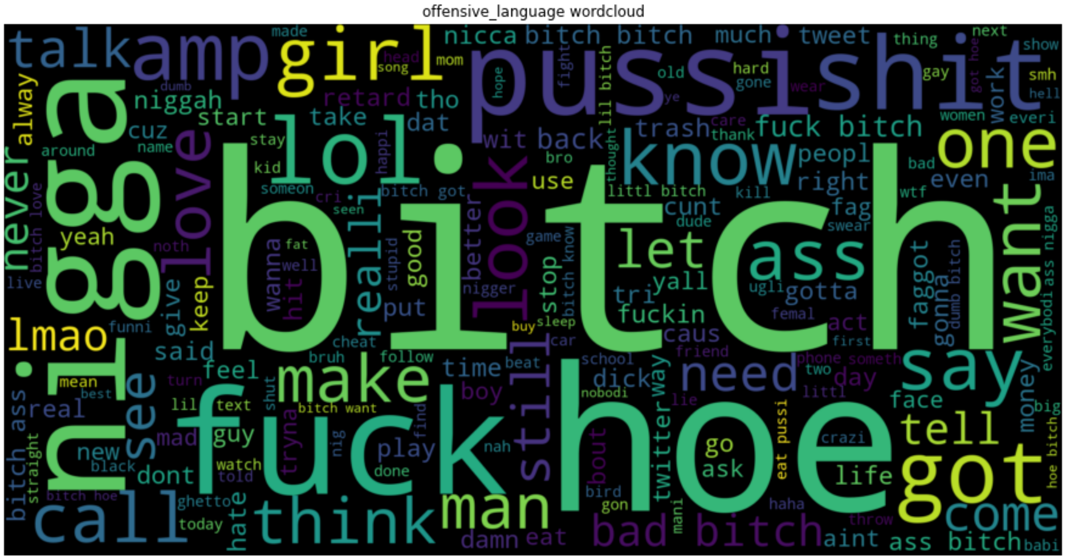


Figure 4.3: Wordcloud of offensive_language tweets

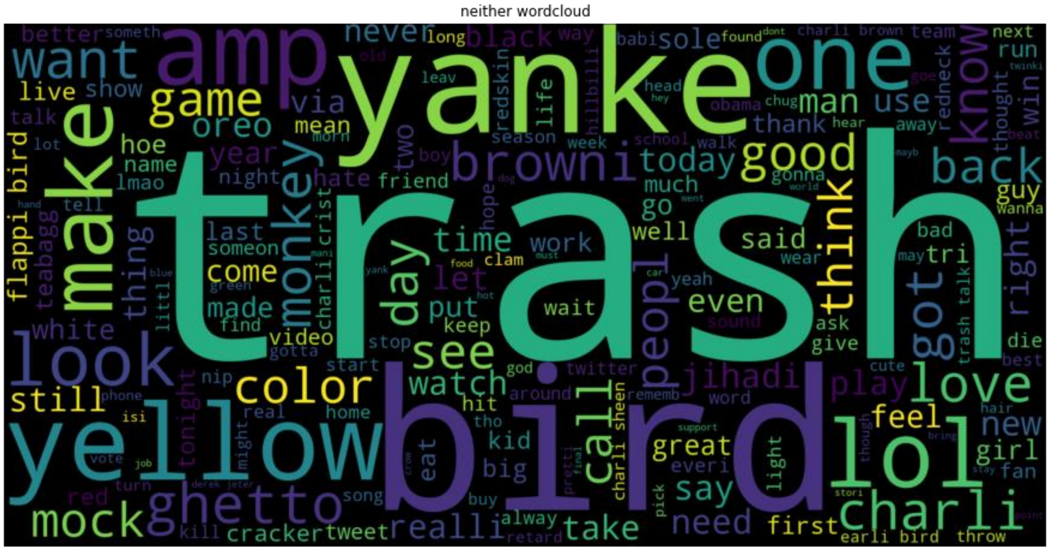


Figure 4.4: Wordcloud of neither tweets

4.5 Modelling:

After data pre-processing and EDA now it's time to get into the nitty-gritty details of modelling. Here we will discuss about the models we used, how we used and what were the hyper-parameters we used to train our models. Our first go to model was Logistic Regression, as it is easy and quick to learn and uses low resources and sometimes gives better results than ensemble methods. Now we needed to consider the imbalance in our dataset and also we needed to test if the model performs better with TF-IDF or without TF-IDF on the tokenized tweets.

4.5.1 Logistic Regression:

So with the very first try we fitted our Logistic Regression model with the actual data which had an imbalanced class and without TF-IDF, the hyper-parameters used were multi_class = ‘ovr’, solver = ‘liblinear’, random_state = 8. In this first iteration the model overlearned about class 1 and predicted most of the test cases as class 1.

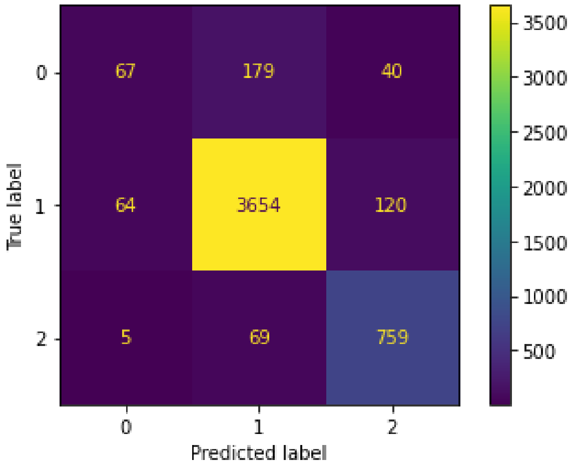


Figure 4.5: Confusion matrix of Logistic Regression

After the first iteration as the data was highly imbalanced we did some balancing and tried undersampling for the second iteration. Now the class 0 - hate_speech had the lowest count of rows at 1430, so in this iteration we randomly choose 1430 each for 2 other classes with the same hyper-parameters. The results were quite better this time than the last time.

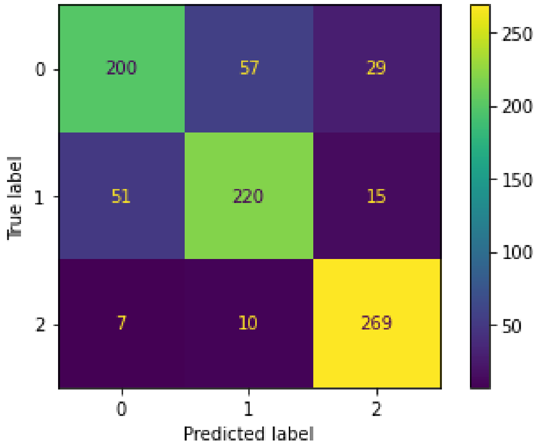


Figure 4.6: Confusion matrix of LR with undersampling

Earlier we tried undersampling now we tried oversampling, as the class 1 - offensive_langauge had most of the rows at 19190 count so we randomly oversampled other 2 classes with the same hyper-parameters. Oversampling produced results better than undersampling, but it was unclear that if the model did find the underlying patterns or it just had overfitting as the data in other 2 classes were mostly repeated. The results weren't pointing to overfitting, so from this point we fed oversampling data to all other models as it performed better than undersampling.

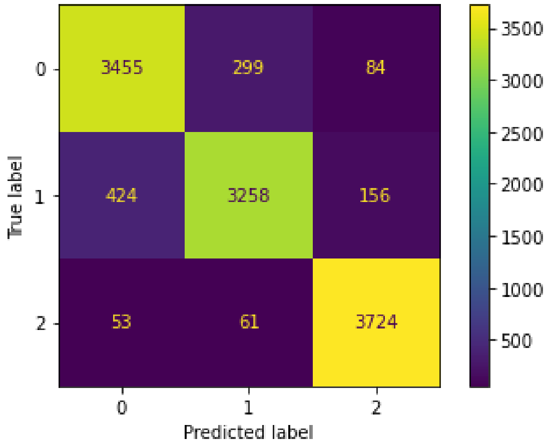


Figure 4.7: Confusion matrix of LR with oversampling

For the final iteration we fed oversampled data with the same hyper-parameters but this time we transformed the tokenized data to TF-IDF and observed the results.

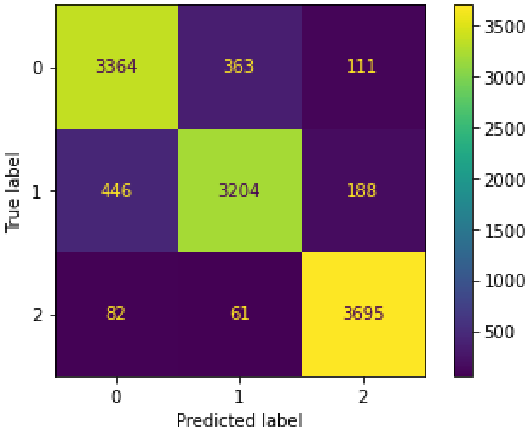


Figure 4.8: Confusion matrix of LR with oversampling and TF-IDF transformation

4.5.2 Naive Bayes:

After we were done with Logistic Regression we moved on with Naive Bayes as it had better results with binary classifications so we taught it to give it a try and see how it performs with our oversampled dataset with and without TF-IDF. Naive Bayes performed well but not as our Logistic Regression model.

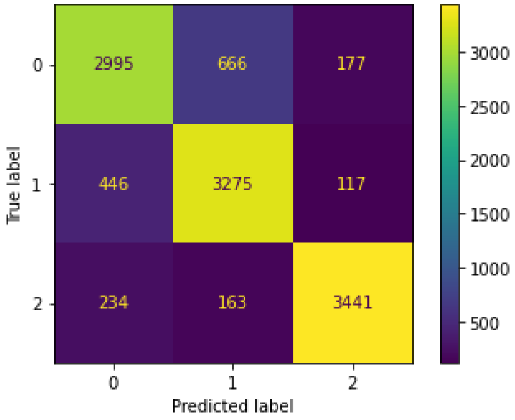


Figure 4.9: Confusion matrix of NB

After this we again trained the model with TF-IDF transformed data to see how it worked.

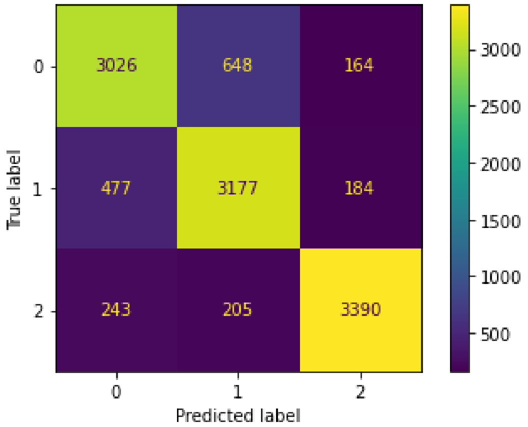


Figure 4.10: Confusion matrix of NB with TF-IDF transformation

4.5.3 Support Vector Machines:

We were curious to use SVM Classifier as it had given state of the art results to (Zampieri et al., 2019a), so first we trained and tested the model with non TF-IDF data, the hyper-parameters used were max_iter = 10000 and class_weight = 'balanced' and found out that it really performed very well better than our Logistic Regression model.

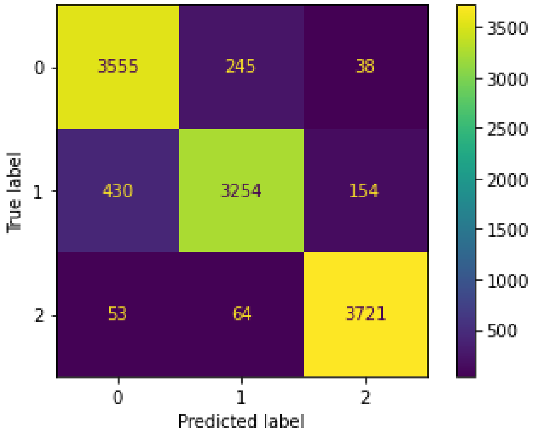


Figure 4.11: Confusion matrix of SVM

After training and testing it with non TF-IDF data we then trained and tested the SVM Classifier with TF-IDF dataset keeping the same hyper-parameters and to our notice we found that the model without the TF-IDF data performed better than the model which was trained with TF-IDF transformed data.

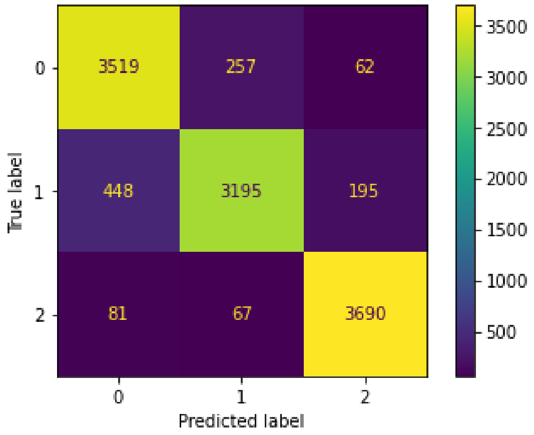


Figure 4.12: Confusion matrix of SVM with TF-IDF transformation

4.5.4 Random Forest:

After simpler models it was time for us to train and test the ensemble methods and see how good they can perform with and without TF-IDF transformed data. The first ensemble method that crossed our mind was Random Forest, so we used Random Forest Classifier with hyper-parameters as n_estimators = 100, criterion = 'entropy', max_depth = 8, min_samples_split = 2.

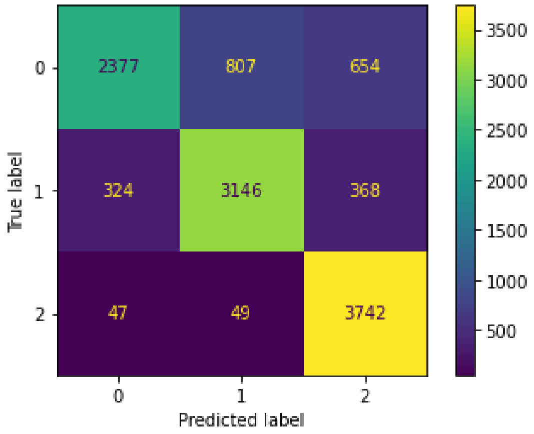


Figure 4.13: Confusion matrix of Random Forest

After this we again trained the model with the TF-IDF dataset and kept the hyper-parameters same.

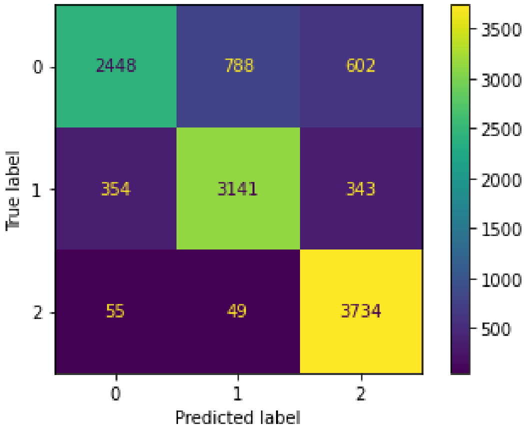


Figure 4.14: Confusion matrix of Random Forest with TF-IDF transformation

4.5.5 XGBoost:

Random Forest Classifier performed decently though we expected it to perform little better, now we had the similar expectations with XGBoost Classifier model so at first we fed the non TF-IDF data and trained the model using hyper-parameters as learning_rate=0.1, max_depth=8, n_estimators=100, eval_metric='auc' and tested the model with it.

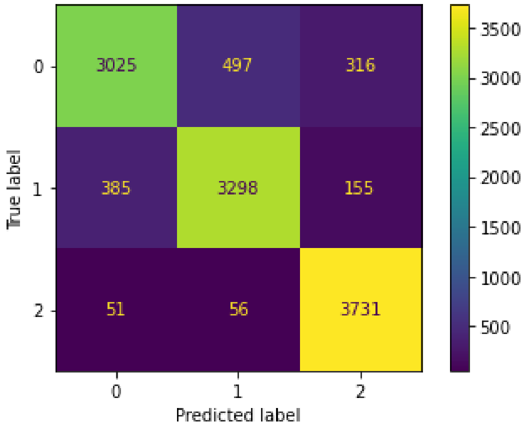


Figure 4.15: Confusion matrix of XGBoost

Once we got the results from XGBoost Classifier without the TF-IDF dataset so this time we trained the model and tested it with the TF-IDF dataset while keeping the hyper-parameters same.

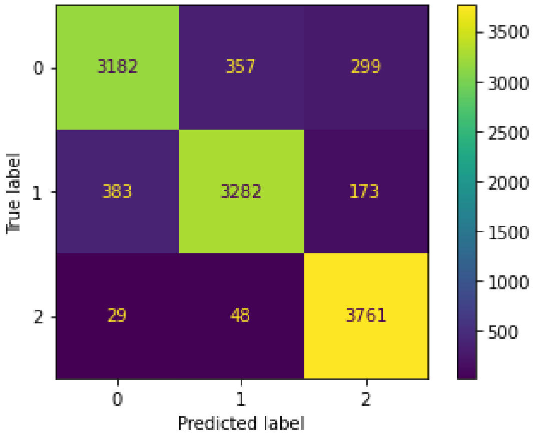


Figure 4.16: Confusion matrix of XGBoost with TF-IDF transformation

4.6 Summary:

In this chapter our major focus was on implementation, we discussed what we implemented and how we implemented. We discussed about the dataset, steps involved in data pre-processing and the models which we implemented and what their hyper-parameters were.

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 Introduction:

Once we are done with the implementation part then it becomes obvious to discuss the results that were derived from the implementations. In this topic we’re going to discuss the results that were derived after training and testing the model with the test or validation dataset.

5.2 Evaluations:

Evaluation is a metric that helps us to understand what works better for us, now the model which works better might also take longer time than the usual or can also take more resources than the others. So we will discuss the results that our models gave and which performed better than the others.

5.2.1 Logistic Regression:

With our first iteration with Logistic Regression which was fed with our original imbalance data we received Accuracy Score: 90.38 %, Precision Score: 75.17 %, Recall Score: 69.92 %, F1 Score: 70.94 %. Now this was imbalance data so we tried to do some balancing with our data, we tried undersampling our data and then trained our model again with the new data and the results were, Accuracy Score: 80.30 %, Precision Score: 80.04 %, Recall Score: 80.30 %, F1 Score: 80.04 %. It was better than the last one but we still needed to see what happens with oversampling so we oversampled our data and fed to Logistic regression model and found out the results were better than the undersampling, we got Accuracy Score: 90.65 %, Precision Score: 90.62 %, Recall Score: 90.65 %, F1 Score: 90.60 %. This was by far the best results we had achieved, but since the data is repeated in oversampling so it's pretty hard to guess if the model actually understood the underlying problem or it just overfitted to some extent. If it was just about accuracy score we could have understood that model might have overfitted precision and recall scores also showed similar progress so it can be understood that model did understand the underlying problem to some extent. Now for our final iteration with the Logistic Regression we first transformed the oversampled data to TF-IDF and then trained to the model on it, the scores we received were, Accuracy Score: 89.13 %, Precision Score: 89.09 %, Recall Score: 89.13 %, F1 Score: 89.07 %. So as we got better results with oversampled data we used this data for further model training and evaluations.

5.2.2 Naive Bayes:

We trained Naive Bayes on the oversampled data first without the TF-IDF and the results were, Accuracy Score: 84.34 %, Precision Score: 84.48 %, Recall Score: 84.34 %, F1 Score: 84.36 %. After this we trained our Naive Bayes model with the TF-IDF transformed data and the results were, Accuracy Score: 83.32 %, Precision Score: 83.43 %, Recall Score: 83.32 %, F1 Score: 83.35 %.

5.2.3 Support Vector Machines:

We also trained Support Vector Machine Classifier with our oversampled data and the results were, Accuracy Score: 91.45 %, Precision Score: 91.49 %, Recall Score: 91.45 %, F1 Score: 91.41 %. Now after this we again trained SVM Classifier with TF-IDF transformed data and the results we got were, Accuracy Score: 90.36 %, Precision Score: 90.40 %, Recall Score: 90.36 %, F1 Score: 90.30 %.

5.2.4 Random Forest:

This was the first ensemble model we tried and first with oversampled data, the results we got from Random Forest Classifier were, Accuracy Score: 80.47 %, Precision Score: 81.22 %, Recall Score: 80.47 %, F1 Score: 79.81 %. The results were good but we needed to train the model with the TF-IDF transformed data, the results we got were, Accuracy Score: 40.79 %, Precision Score: 71.92 %, Recall Score: 40.79 %, F1 Score: 31.09 %.

5.2.5 XGBoost:

After Random Forest we had one more ensemble model which is well reputed, it was XGBoost Classifier so we trained the model with the oversampled data and the results we got were, Accuracy Score: 87.32 %, Precision Score: 87.28 %, Recall Score: 87.32 %, F1 Score: 87.16 %. We again trained the XGBoost Classifier model with the TF-IDF transformed data to check it's performance, the results we got were, Accuracy Score: 41.72 %, Precision Score: 69.12 %, Recall Score: 41.72 %, F1 Score: 32.68 %.

5.3 Summary:

We discussed the results we got from each model with the TF-IDF transformed and non-transformed data and how well these models performed with the data, first we tried simpler models and then moved on to the sophisticated ensemble models and how well they performed. Let's have a look at the results in a tabular format:

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Logistic Regression	90.38	75.17	69.92	70.94
Logistic Regression (undersampling)	80.30	80.04	80.30	80.04
Logistic Regression (oversampling)	90.65	90.62	90.65	90.60
Logistic Regression (TF-IDF)	89.13	89.09	89.13	89.07
Naïve Bayes	84.34	84.48	84.34	84.36
Naïve Bayes (TF-IDF)	83.32	83.43	83.32	83.35
Support Vector Machine	91.45	91.49	91.45	91.41
Support Vector Machine (TF-IDF)	90.36	90.40	90.36	90.30
Random Forest	80.47	81.22	80.47	79.81
Random Forest (TF-IDF)	40.79	71.92	40.79	31.09
XGBoost	87.32	87.28	87.32	87.16
XGBoost (TF-IDF)	41.72	69.12	41.72	32.68

Table 1: Results table

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

6.1 Introduction:

Once we did pre-processing and trained our models to the data and got the results from them, then it comes to learning and conclusion, so in this section, we’ll discuss the conclusion that we have reached looking at the results of how well our models performed. We’ll recommend it to anyone who is doing further research in this field.

6.2 Discussions and Conclusion:

After going through all these pre-processing processes and model training and validation we can conclude that the Support Vector Machine Classifier works better than Logistic Regression, Naive Bayes, Random Forest Classifier and XGBoost Classifier in the field of Abusive Language Detection with three classes i.e. hate_speech, offensive_laguage and neither. Though one might have high hopes with ensemble models like Random Forest or XGBoost, SVM performs better than these.

Also, we can conclude that the TF-IDF transformation didn’t serve a benefit but rather it caters for loss and in contexts of ensemble models, they crumble to their feet with TF-IDF transformation, their scores drastically falls downs, while other models performed a lot better with very little loss.

6.3 Future Recommendations:

In this study, we used Logistic Regression, Naive Bayes, Support Vector Machine Classifier, Random Forest Classifier and XGBoost Classifier, we didn’t use any of the Hugging Face BERT models or Deep Learning models, so there a clear opportunity to use those model and see how well they perform on the dataset also with the time we also need to change the dataset to newer dataset as people will pull new tricks from their bag so from time to time we need to retrain our models with the newer dataset which contains newer slangs.

6.4 Summary:

We discussed the conclusion from our study that we did in the field of Abusive Language Detection in NLP and also recommended future recommendations to anyone who will be doing further research in the future in this field.

REFERENCES

- Ali, R., Farooq, U., Arshad, U., Shahzad, W. and Beg, M.O., (2022) Hate speech detection on Twitter using transfer learning. *Computer Speech & Language*, 74, p.101365.

- Alrehili, A., (2019) Automatic Hate Speech Detection on Social Media: A Brief Survey. In: *2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*. [online] 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA). Abu Dhabi, United Arab Emirates: IEEE, pp.1–6. Available at: https://ieeexplore.ieee.org/document/9035228/ [Accessed 14 Sep. 2022].

- Alsafari, S., Sadaoui, S. and Mouhoub, M., (2020) Hate and offensive speech detection on Arabic social media. *Online Social Networks and Media*, 19, p.100096.

- Anand, M., Sahay, K.B., Ahmed, M.A., Sultan, D., Chandan, R.R. and Singh, B., (2022) Deep learning and natural language processing in computation for offensive language detection in online social networks by feature selection and ensemble classification techniques. *Theoretical Computer Science*, p.S0304397522003887.

- Arango, A., Pérez, J. and Poblete, B., (2022) Hate speech detection is not as easy as you may think: A closer look at model validation (extended version). *Information Systems*, 105, p.101584.

- Ayo, F.E., Folorunso, O., Ibharalu, F.T. and Osinuga, I.A., (2020) Machine learning techniques for hate speech classification of twitter data: State-of-the-art, future challenges and research directions. *Computer Science Review*, 38, p.100311.

- Ayo, F.E., Folorunso, O., Ibharalu, F.T., Osinuga, I.A. and Abayomi-Alli, A., (2021) A probabilistic clustering model for hate speech classification in twitter. *Expert Systems* *with Applications*, 173, p.114762.

- Badjatiya, P., Gupta, S., Gupta, M. and Varma, V., (2017) Deep Learning for Hate Speech Detection in Tweets. In: *Proceedings of the 26th International Conference on World Wide Web Companion - WWW ’17 Companion*. [online] the 26th International Conference. Perth, Australia: ACM Press, pp.759–760. Available at: http://dl.acm.org/citation.cfm?doid=3041021.3054223 [Accessed 14 Sep. 2022].

- Biradar, S., Saumya, S. and Chauhan, A., (2021) Hate or Non-hate: Translation based hate speech identification in Code-Mixed Hinglish data set. In: *2021 IEEE International Conference on Big Data (Big Data)*. [online] 2021 IEEE International Conference on Big Data (Big Data). Orlando, FL, USA: IEEE, pp.2470–2475. Available at: https://ieeexplore.ieee.org/document/9671526/ [Accessed 14 Sep. 2022].

- Charitidis, P., Doropoulos, S., Vologiannidis, S., Papastergiou, I. and Karakeva, S., (2020) Towards countering hate speech against journalists on social media. *Online Social Networks and Media*, 17, p.100071.

- Davidson, T., Warmsley, D., Macy, M. and Weber, I., (2017) *Automated Hate Speech Detection and the Problem of Offensive Language*. Available at: http://arxiv.org/abs/1703.04009 [Accessed 14 Sep. 2022].

- Djuric, N., Zhou, J., Morris, R., Grbovic, M., Radosavljevic, V. and Bhamidipati, N., (2015) Hate Speech Detection with Comment Embeddings. In: *Proceedings of the 24th International Conference on World Wide Web*. [online] WWW ’15: 24th International World Wide Web Conference. Florence Italy: ACM, pp.29–30. Available at: https://dl.acm.org/doi/10.1145/2740908.2742760 [Accessed 14 Sep. 2022].

- El-Alami, F., Ouatik El Alaoui, S. and En Nahnahi, N., (2022) A multilingual offensive language detection method based on transfer learning from transformer fine-tuning model. *Journal of King Saud University - Computer and Information Sciences*, 348, pp.6048–6056.

- ElSherief, M., Kulkarni, V., Nguyen, D., Wang, W.Y. and Belding, E., (2018) *Hate Lingo: A Target-based Linguistic Analysis of Hate Speech in Social Media*. Available at: http://arxiv.org/abs/1804.04257 [Accessed 14 Sep. 2022].

- Fortuna, P. and Nunes, S., (2019) A Survey on Automatic Detection of Hate Speech in Text. *ACM Computing Surveys*, 514, pp.1–30.

- Fortuna, P., Soler-Company, J. and Wanner, L., (2021) How well do hate speech, toxicity, abusive and offensive language classification models generalize across datasets? *Information Processing & Management*, 583, p.102524.

- Gambäck, B. and Sikdar, U.K., (2017) Using Convolutional Neural Networks to Classify Hate-Speech. In: *Proceedings of the First Workshop on Abusive Language Online*. [online] Proceedings of the First Workshop on Abusive Language Online. Vancouver, BC, Canada: Association for Computational Linguistics, pp.85–90. Available at: http://aclweb.org/anthology/W17-3013 [Accessed 14 Sep. 2022].

- Jha, V.K., P, H., P N, V., Vijayan, V. and P, P., (2020) DHOT-Repository and Classification of Offensive Tweets in the Hindi Language. *Procedia Computer Science*, 171, pp.2324–2333.

- Kaur, S., Singh, S. and Kaushal, S., (2021) Abusive Content Detection in Online User-Generated Data: A survey. *Procedia Computer Science*, 189, pp.274–281.

- Khan, M.U.S., Abbas, A., Rehman, A. and Nawaz, R., (2021) HateClassify: A Service Framework for Hate Speech Identification on Social Media. *IEEE Internet Computing*, 251, pp.40–49.

- Lee, H.-S., Lee, H.-R., Park, J.-U. and Han, Y.-S., (2018) An abusive text detection system based on enhanced abusive and non-abusive word lists. *Decision Support Systems*, 113, pp.22–31.

- Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y. and Chang, Y., (2016) Abusive Language Detection in Online User Content. In: *Proceedings of the 25th International Conference on World Wide Web*. [online] WWW ’16: 25th International World Wide Web Conference. Montréal Québec Canada: International World Wide Web Conferences Steering Committee, pp.145–153. Available at: https://dl.acm.org/doi/10.1145/2872427.2883062 [Accessed 14 Sep. 2022].

- Ombui, E., Muchemi, L., Wagacha, P., Gichamba, A. and Karani, M., (2019) Leveraging Hierarchical Features for HateSpeech Identification in Short Message Texts. In: *2019 IEEE AFRICON*. [online] 2019 IEEE AFRICON. Accra, Ghana: IEEE, pp.1–5. Available at: https://ieeexplore.ieee.org/document/9133781/ [Accessed 14 Sep. 2022].

- Pamungkas, E.W., Basile, V. and Patti, V., (2021) A joint learning approach with knowledge injection for zero-shot cross-lingual hate speech detection. *Information Processing & Management*, 584, p.102544.

- Park, J.H. and Fung, P., (2017) One-step and Two-step Classification for Abusive Language Detection on Twitter. In: *Proceedings of the First Workshop on Abusive Language Online*. [online] Proceedings of the First Workshop on Abusive Language Online. Vancouver, BC, Canada: Association for Computational Linguistics, pp.41–45. Available at: http://aclweb.org/anthology/W17-3006 [Accessed 7 Sep. 2022].

- Schmidt, A. and Wiegand, M., (2017) A Survey on Hate Speech Detection using Natural Language Processing. In: *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*. [online] Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media. Valencia, Spain: Association for Computational Linguistics, pp.1–10. Available at:

http://aclweb.org/anthology/W17-1101 [Accessed 14 Sep. 2022].

- Sharma, A., Kabra, A. and Jain, M., (2022) Ceasing hate with MoH: Hate Speech Detection in Hindi–English code-switched language. *Information Processing & Management*, 591, p.102760.

- Silva, L., Mondal, M., Correa, D., Benevenuto, F. and Weber, I., (2016) *Analyzing the Targets of Hate in Online Social Media*. Available at: http://arxiv.org/abs/1603.07709 [Accessed 14 Sep. 2022].

- Song, R., Giunchiglia, F., Shen, Q., Li, N. and Xu, H., (2022) Improving Abusive Language Detection with online interaction network. *Information Processing & Management*, 595, p.103009.

- Subramanian, M., Ponnusamy, R., Benhur, S., Shanmugavadivel, K., Ganesan, A., Ravi, D., Shanmugasundaram, G.K., Priyadharshini, R. and Chakravarthi, B.R., (2022) Offensive language detection in Tamil YouTube comments by adapters and cross-domain knowledge transfer. *Computer Speech & Language*, 76, p.101404.

- Wadud, Md.A.H., Kabir, M.M., Mridha, M.F., Ali, M.A., Hamid, Md.A. and Monowar, M.M., (2022) How can we manage Offensive Text in Social Media - A Text Classification Approach using LSTM-BOOST. *International Journal of Information Management Data Insights*, 22, p.100095.

- Waseem, Z., Davidson, T., Warmsley, D. and Weber, I., (2017) Understanding Abuse: A Typology of Abusive Language Detection Subtasks. In: *Proceedings of the First Workshop on Abusive Language Online*. [online] Proceedings of the First Workshop on Abusive Language Online. Vancouver, BC, Canada: Association for Computational Linguistics, pp.78–84. Available at: http://aclweb.org/anthology/W17-3012 [Accessed 14 Sep. 2022].

- Waseem, Z. and Hovy, D., (2016) Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. In: *Proceedings of the NAACL Student Research Workshop*. [online] Proceedings of the NAACL Student Research Workshop. San Diego, California: Association for Computational Linguistics, pp.88–93. Available at: http://aclweb.org/anthology/N16-2013 [Accessed 14 Sep. 2022].

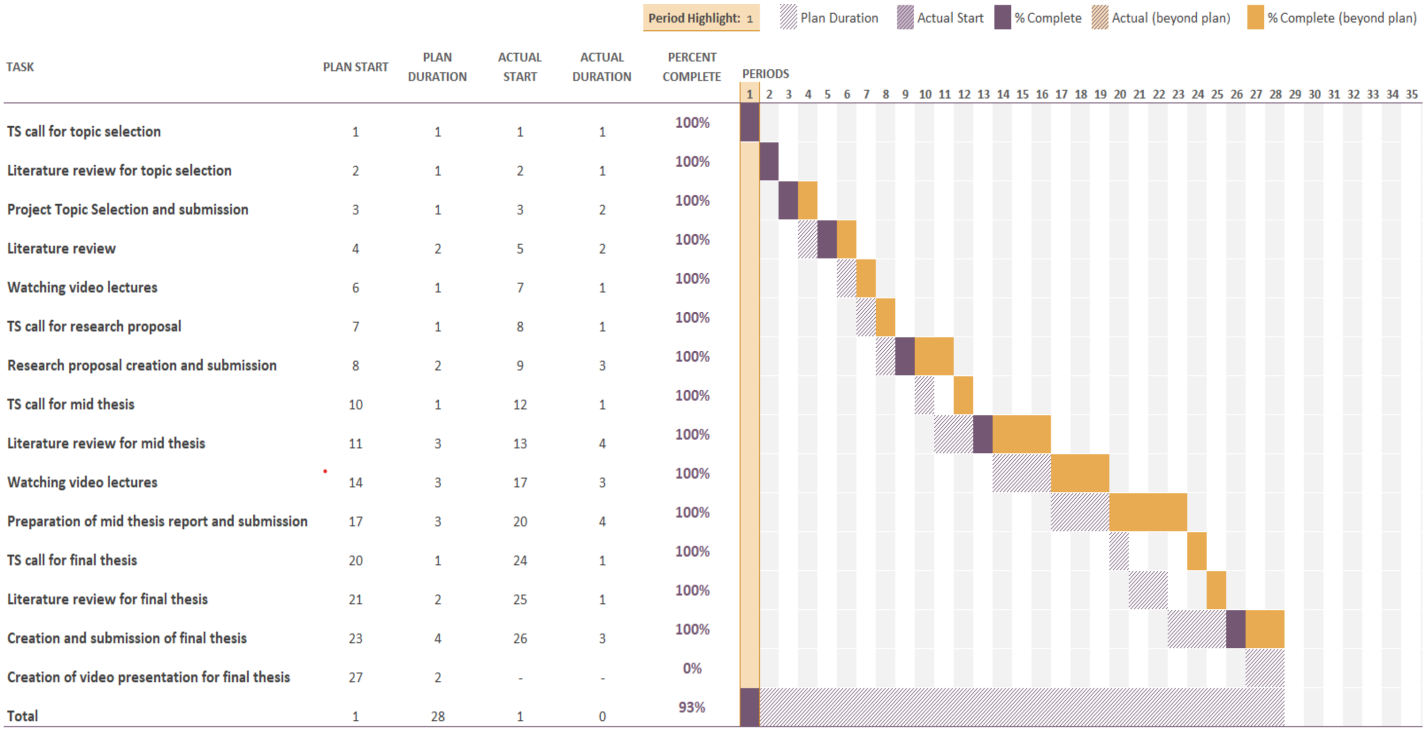
- Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N. and Kumar, R., (2019a) Predicting the Type and Target of Offensive Posts in Social Media. In: *Proceedings of the 2019 Conference of the North*. [online] Proceedings of the 2019 Conference of the North. Minneapolis, Minnesota: Association for Computational Linguistics, pp.1415–1420. Available at: http://aclweb.org/anthology/N19-1144 [Accessed 14 Sep. 2022].

- Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N. and Kumar, R., (2019b) SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval). In: *Proceedings of the 13th International Workshop on Semantic Evaluation*. [online] Proceedings of the 13th International Workshop on Semantic Evaluation. Minneapolis, Minnesota, USA: Association for Computational Linguistics, pp.75–86. Available at: https://www.aclweb.org/anthology/S19-2010 [Accessed 14 Sep. 2022].

- Zhang, Z. and Luo, L., (2019) Hate speech detection: A solved problem? The challenging case of long tail on Twitter. *Semantic Web*, 105, pp.925–945.

- Zhang, Z., Robinson, D. and Tepper, J., (2018) Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network. In: A. Gangemi, R. Navigli, M.-E. Vidal, P. Hitzler, R. Troncy, L. Hollink, A. Tordai and M. Alam, eds., *The Semantic Web*, Lecture Notes in Computer Science. [online] Cham: Springer International Publishing, pp.745–760. Available at: http://link.springer.com/10.1007/978-3-319-93417-4_48 [Accessed 14 Sep. 2022].

APPENDIX A: RESEARCH PLAN



TASK	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
TS call for topic selection	1	1	1	1	100%
Literature review for topic selection	2	1	2	1	100%
Project Topic Selection and submission	3	1	3	2	100%
Literature review	4	2	5	2	100%
Watching video lectures	6	1	7	1	100%
TS call for research proposal	7	1	8	1	100%
Research proposal creation and submission	8	2	9	3	100%
TS call for mid thesis	10	1	12	1	100%
Literature review for mid thesis	11	3	13	4	100%
Watching video lectures	14	3	17	3	100%
Preparation of mid thesis report and submission	17	3	20	4	100%
TS call for final thesis	20	1	24	1	100%
Literature review for final thesis	21	2	25	1	100%
Creation and submission of final thesis	23	4	26	3	100%
Creation of video presentation for final thesis	27	2	-	-	0%
Total	1	28	1	0	93%

Figure A: Research Plan

APPENDIX B: RESEARCH PROPOSAL

ABUSIVE LANGUAGE DETECTION ON SOCIAL MEDIA

RAJKUMAR RAJENDRAPRASAD PAL

RESEARCH PROPOSAL

AUGUST 2022

Abstract

Since internet access has reached to everyone in their hands through mobile phones we have moved into tech era or internet era where internet has become the basic necessity of livelihood, with the internet social media has also grown humongous, so much so that majority of internet usage is the part of social media usage. Just as selfies, dog and cat pics and other good things abusive and offensive content has also made their way to social media. Social media provides a forum for people to communicate their opinions, expertise, experiences, and feelings, but it becomes a serious issue when these interactions are used as a vehicle for abusive statements, comments, and dialogues. Racism, sexism, and other ideologies may all be promoted through the use of abusive language. Here in this research we will be working dataset with tweets, twitter is a microblogging platform that empowers users to communicate with others through their tweets which are of 140 characters, hence it produces large amounts of data from brief digital material.

Computational techniques are one of the best approaches to separate unwanted information. Using machine learning algorithms to separate abusive content from the ones which aren't is one of the best ways to handle this problem. Although the creation of a general metadata architecture has received minimal attention from researchers, abusive text classification on a Twitter dataset has remained a hot topic of study. The automated detection of hate speech and associated phenomena is the subject of a sizable corpus of study.

In this study we try to differentiate the tweets which contains abusive words from the one’s which aren’t so that the algorithms can be further used to clear the tweets from the content wall of the users to whom the abusive content should be hidden. We have used some simple algorithms to complex ensemble algorithms for or abusive content detection on our available data.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

1. ML:	Machine Learning
2. SVM:	Support Vector Machine
3. RLC:	Rocchio Linear Classifier
4. MTC:	Multilingual Text Categorization
5. KNN:	K-Nearest Neighbour
6. SOM:	Self Organising Maps
7. GA:	Genetic Algorithm
8. CF:	CrowdFlower
9. Mbps:	Megabits Per Second
10. NLTK:	Natural Language Toolkit
11. SciKit:	Scikit-learn

Table of Contents

1. Background

In this modern era as the time progresses, we’re moving into the digital era. Now internet has become more important than food, people can live a day without food but not internet. With increase in usage of internet crime, abusiveness, bullying and hate speech has also increased over internet (Ayo et al., 2020) and a parents wouldn’t want their children to learn this from internet which they will eventually learn truth be told, but not before time.

Talking about internet, the majority of the internet usage is over social media and bullying, abusiveness and hate speech has also found their way to social media, (Fortuna et al., 2021) people these days are trying to fast to reduce their intake of social media, but eventually if there’s abusiveness we will take it deep down on our subconscious mind, and if fed to young child imagine what future we have when these children become adult and have their own children. (Ayo et al., 2021)

So in this research work I’m trying to find the abusive language or word on the social media, which can be further used to prevent it from being fed a young child or even from elderly people, in short it will be a small part in a bigger picture to prevent abusiveness from social media and trying to make the clean social media experience. (Biradar et al., 2021; Khan et al., 2021; Ali et al., 2022)

2. Problem Statement OR Related Research OR Related Work

Text classification and abusive language detection are not something new, there are methods involving doing several monolingual operations on datasets in Chinese and English. In a different study, the author puts out a system for categorizing works in which contents were translated into a common tongue. To categorize text, they employed the RLC, Naive Bayes, and KNN after using WordNet to link terms to concepts. The co-regularization and consensus-based self-training methods that the author combined to study MTC doesn’t include translation but instead looked towards the WordNet associated with every language (Anand et al., 2022). Using N-gram techniques, the author tackled multilingual text classification. Spanish, Italian, and English were the three languages they studied at MTC. (El-Alami et al., 2021) They began by making educated guesses about the study language before categorizing it with Naive Bayes. A host of ML techniques, including SVM, Decision Trees, KNN, SOM, and GA, have more recently been utilized to address the MTC problem in both Hindi and English. To increase the accuracy of the procedure, respective authors employed several feature selection techniques. The effectiveness of a pattern-based method was compared to that of KNN, Random Forest, and SVM to identify sarcasm in tweets. The majority of these methods rely on pre-trained text classification models (Anand et al., 2022). The great bulk of research on this topic focuses on English, in part because English-language resources are more widely available. Recently studies were done on different languages including Greek, Arabic, Dutch, Danish, French, Portuguese, Italian, Turkish and Slovene which lead us to additional datasets and tools for the above languages which can be further used to narrow down research on a particular language of choice. In context with abusive language detection online, two questions were addressed viz., 1. Which is more important for cross-dataset generalisation, the models or the datasets? 2. Additionally, which model and dataset properties are crucial for generalisation at the finally? (Fortuna et al., 2021). After this research there were still few work that is remained, potential area of research is the use of combined datasets with the use of a category conversion schema prior to the merging, allowing for a more fine-grained categorization. Recent works have shown high performance on merged datasets. (Jha et al., 2020; Fortuna et al., 2021; Khan et al., 2021; Arango et al., 2022)

3. Research Questions (If any)

The principal question we ask ourselves while doing our research is to find that "Does the current tweet contains any abusive language?"

4. Aim and Objectives

The foremost aim of this research or study is to propose and compare models that can do the job of finding abusive word or content in the given tweet and or social media comment.
The objectives of this research can be found by diving the aim into further steps:

- The foremost part is to gather a dataset of tweets or social media comments and cleanse it to make suitable for our models.
- Predict the statement if it has any abusive words in it.
- Compare and evaluate the results of multiple models.

5. Significance of the Study

The significance of this research is to contribute to the social media to make it a safer and cleaner environment for kids and elderly people, so when they visit social media they would not be bombarded with the tweets which contains abusive language that might upset the mood of the kid or elderly person or worst case teach them to use those abusive language and spread it in real world.

6. Scope of the Study

The following things are inside the scope of this study or research:

- To find the abusive word in a tweet
- To suggest the better model based on its performance on dataset

The following things are not inside the scope of this study or research:

- To replace the abusive word with another word or * or #.
- To replace the sentence or tweet to something new with similar meaning.
- To suggest the tweets without abusive words in them.

7. Research Methodology

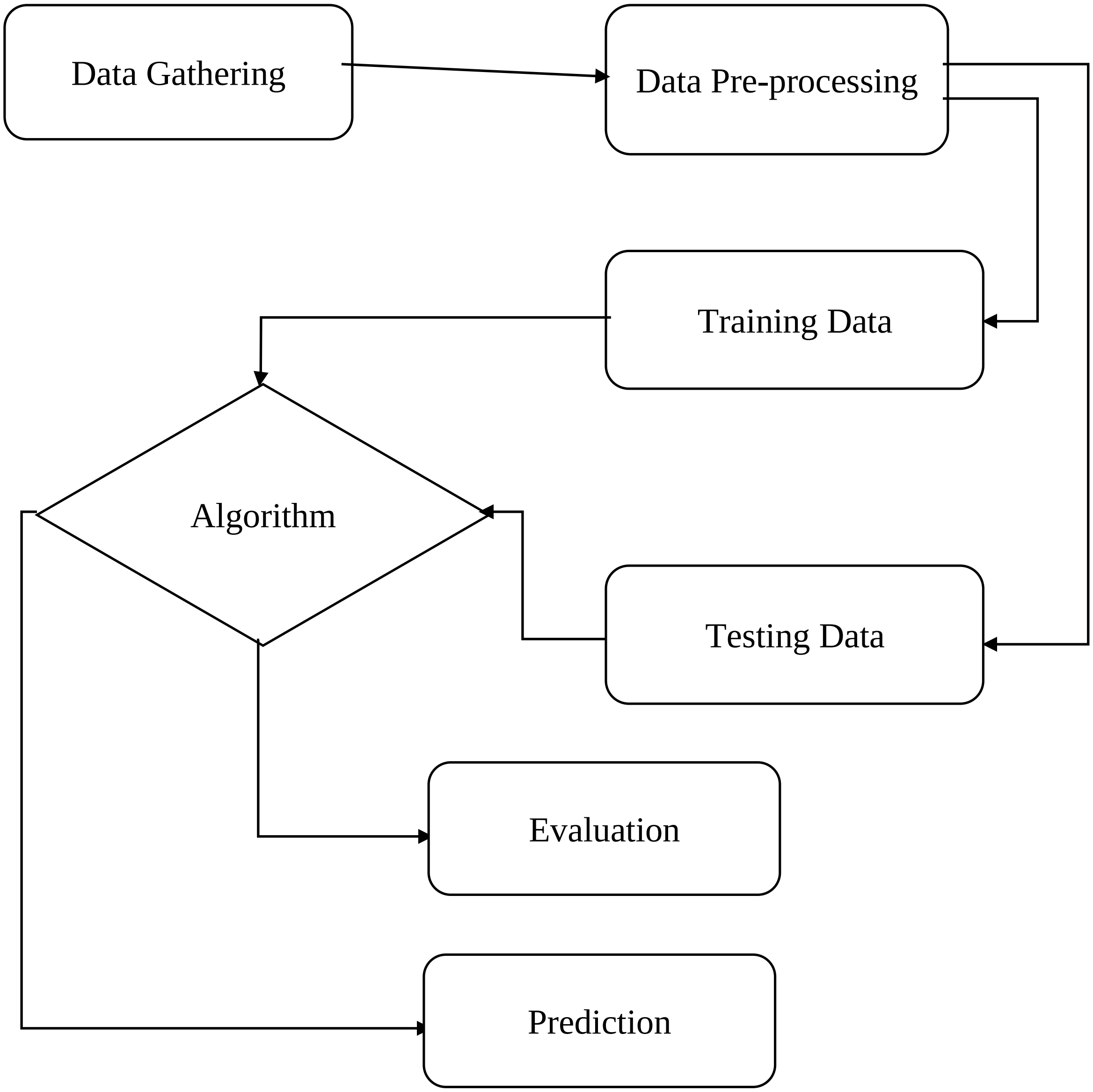
7.1. Dataset:

The dataset which we are using here is the twitter Davidson dataset, it is found on Repository for the paper "Automated Hate Speech Detection and the Problem of Offensive Language", ICWSM 2017. It is an English dataset with all the tweets in English language.

The dataset contains 24783 tweets, each tweet is either of hate speech or abusive language or neither as voted by the users of CrowdFlower.

The dataset contains 5 columns and they are:

1. count: “amount of individuals that coded each tweet on CrowdFlower (min is 3).” hate_speech: “amount of users on CF who voted the tweet to be a hate speech.”
2. offensive_language: “amount of users on CF who voted the tweet to be offensive.”
3. neither: “amount of users on CF who voted the tweet to be neither non – offensive nor offensive.”
4. class: “class label for majority of users on CF. 0 - hate speech 1 - offensive language 2 – neither.”



7.2. Pre-processing:

Data cleaning or pre-processing is essential as the original dataset is usually noisy and poorly organised on top of it, it might also have data that might be irrelevant to our objective. The dataset contains lots of data that is of no use to us, i.e. our objective is to find abusive word in the tweets, so we will remove the columns that are not associated with offensive or abusive language. Once we have remove the columns that are not required the next step would be to convert all the tweets into small case and after conversion removal of unnecessary words that are irrelevant to our research i.e. removal of punctuations, user handles or mentions, hast tags (#) and emoticons. After this step we need tokenize and normalize our tweets. Next step would be de-duplication, this step would be helpful to identify the duplicate tweets that might influence our ML model to be biased. If the dataset is skewed we need to log transformation and if the dataset is imbalanced then we need to do resampling.

7.3. Models:

Our approach for prediction will be to start from bottom and move upwards i.e. to start from basic model and then move on to complex models. So we will start our classification from Logistic Regression and Decision trees once we have some results we can move on to SVM or Random Forest and see how they are preforming on our train and test dataset. In our study the main objective is binary classification, so Naïve Bayes serves better with binary classification. Once that is done we can then finally move on to complex XGBoost Classifier. In Random Forest and XGBoost we will try hyper parameter tunings that can give us different results and train our model to perform better.

7.4. Evaluation:

We can see that selecting the appropriate model is also crucially important. Testing models on a dataset provides important information on the quality of dataset in a nascent research field like abusive language detection, where so much subjectivity still remains. The evaluation metrics we will use in our study to evaluate the performance of our models are:

- Accuracy: “The probability of correctly classified abusive tweets from the total number of tweets present.”

- Precision: “It is the probability of the predicted abusive tweets that were actually labelled as abusive.”
- Recall: “It is the probability of abusive tweets correctly predicted as abusive tweets.”

8. Requirements Resources

Hardware requirements:

- A desktop computer or laptop with Intel Core i3 5th gen or AMD FX series.
- 8GB RAM

Software requirements:

- Python 3.x or above
- NLTK libraries
- SciKit library

Other requirements:

- Internet connectivity with 2Mbps or higher

9. Research Plan

Task	Completed (%)	Planned Duration(weeks)	Actual Duration (weeks)
TS call for topic selection	100	1	1
Literature review for topic selection	100	1	1
Project Topic Selection and submission	100	1	2
Literature review	100	2	2

Watching recorded lectures	100	1	1
TS call for research proposal	100	1	1
Research proposal creation and submission	100	2	3
TS call for mid thesis	0	1	—
Literature review for mid thesis	0	3	—
Watching recorded lectures	0	2	—
Preparation of mid thesis report and submission	0	3	—
TS call for final thesis	0	1	—
Literature review for final thesis	0	2	—
Creation and submission of final thesis	0	4	—
Creation of PowerPoint presentation for final thesis	0	2	—

References

- Ali, R., Farooq, U., Arshad, U., Shahzad, W. and Beg, M.O., (2022) Hate speech detection on Twitter using transfer learning. *Computer Speech & Language*, 74, p.101365.

- Anand, M., Sahay, K.B., Ahmed, M.A., Sultan, D., Chandan, R.R. and Singh, B., (2022) Deep learning and natural language processing in computation for offensive language detection in online social networks by feature selection and ensemble classification techniques. *Theoretical Computer Science*.

- Arango, A., Pérez, J. and Poblete, B., (2022) Hate speech detection is not as easy as you may think: A closer look at model validation (extended version). *Information Systems*, 105, p.101584.

- Ayo, F.E., Folorunso, O., Ibharalu, F.T. and Osinuga, I.A., (2020) Machine learning techniques for hate speech classification of twitter data: State-of-the-art, future challenges and research directions. *Computer Science Review*, 38, p.100311.

- Ayo, F.E., Folorunso, O., Ibharalu, F.T., Osinuga, I.A. and Abayomi-Alli, A., (2021) A probabilistic clustering model for hate speech classification in twitter. *Expert Systems with Applications*, 173, p.114762.

- Biradar, S., Saumya, S. and Chauhan, A., (2021) Hate or Non-hate: Translation based hate speech identification in Code-Mixed Hinglish data set. In: *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, pp.2470–2475.

- El-Alami, F., Ouatik El Alaoui, S. and En Nahnahi, N., (2021) A multilingual offensive language detection method based on transfer learning from transformer fine-tuning model. *Journal of King Saud University - Computer and Information Sciences*.

- Fortuna, P., Soler-Company, J. and Wanner, L., (2021) How well do hate speech, toxicity, abusive and offensive language classification models generalize across datasets? *Information Processing & Management*, [online] 583, p.102524. Available at: https://linkinghub.elsevier.com/retrieve/pii/S0306457321000339.

- Jha, V.K., P, H., P N, V., Vijayan, V. and P, P., (2020) DHOT-Repository and Classification of Offensive Tweets in the Hindi Language. *Procedia Computer Science*, 171, pp.2324–2333.

- Khan, M.U.S., Abbas, A., Rehman, A. and Nawaz, R., (2021) HateClassify: A Service Framework for Hate Speech Identification on Social Media. *IEEE Internet Computing*, 251, pp.40–49.