Detecting Hate Speech in Social Media Articles in Romanized Sinhala

A dissertation submitted for the Degree of Master of Computer Science

N.W.Hettiarachchi
University of Colombo School of Computing
2020

Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: N.W.Hettiarachchi

Registration Number: 2017/MCS/037

Index Number: 17440372

_____

Signature: Date:

This is to certify that this thesis is based on the work of

Mr. /Ms. N.W.Hettiarachchi

Under my supervision. The thesis has been prepared according to the format stipulated and is of

Acceptable standard.

Certified by:

Dr. A. R. Weerasinghe

_____

Signature: Date:

Abstract

The main aim of this research is to automatically identify the hate content of social media comments and documents written by the Romanized Sinhala Language. Also most of researched done the hate speech recognition study in English or their language but here try to identify the hate speech in Romanized Sinhala language.

Hate words and other hated texts are growing issue, and to combat this they turn to machine learning and computer science.

In this research compare the several features extraction methods and four machine learning algorithms for hate speech detection Also compare difference N-gram values such as unigram, bigram and trigram and used the value of Min-Df as 3. This study will investigate and compare different features for the different classifier when classifying hate speech comments on Facebook. We have achieved a data set of nearly 2500 comments, some containing hate speech, and trained and tested our classifier with different features and finally examine the Multinomial Naive Bayes Classifier is performed better than other classification models Also compare the feature extraction methods countvectorizer and TfIdfVectorizer, we examined all the best performing models is TfIdf Vectorizer.

In the random forest classifier method, when we evaluating those results we can see some overfitting the result on that classification methods. So used the parameter tuning for the all classification algorithms especially for the random forest classifier change the n_estimators value and random_state value then can see the some best results. According to the above examined of Final Results of Tf-idf Vectorizer feature extraction method the Multinomial Naive Bayes Classifier model is better than other models with bigram and min _Df value is 3. Multinomial Naive Bayes Classifier result with bigram and min _Df value is 3.

Acknowledgement

First and foremost, My heartfelt thanks to my supervisor, Dr. A.R Weerasinghe, Senior Lecturer at the University of Colombo School of Computing (UCSC), for giving me the opportunity to complete research. And would like to express sincere thanks. Continued support for my final year research in the master of Computer Science Degree. I appreciate his time and comments on making my work productive and exciting. His valuable suggestions, ideas, and guidance encouraged me to do this research. His deep understanding of this research area helped me to succeed in this research. I am indebted to him for his generosity, unselfish support, and especially for the excellent example and patience that he gave me this year. Big thanks go to him again, because without him this work would not be successful.

I also thank our MCS Project Coordinator as well as my co-supervisor Dr.Randil Pushpananda senior lecturer of the University Of Colombo School Of Computing – UCSC for his guidance has been given throughout the year.

Finally, I would like to express my gratitude to my parents, siblings, and friends from my Msc team for providing me with unshakable support and ongoing support throughout the research process. Without them, this research would not have been possible. Thank You.

Table of Contents

List of Figures ..... **vii**

List of Tables ..... **vii**

List of Acronyms ..... **xi**

Chapter 1 – Introduction **1**

1.1 Background to the Research 1

1.1.1 What is social media? 1

1.1.2 Social media and hate Speech 2

1.2 Definition for Hate Speech 3

1.3 Problem Statement 3

1.4 Research Problem 4

1.5 Motivation 4

1.6 Research Contribution 4

1.6.1 Goal 4

1.6.2 Objectives 5

1.7 Scope and Limitations 5

1.8 Justification for the Research 6

1.9 Structure of the Dissertation 6

Chapter 2: Literature review **7**

2.1 Lexical based approach 7

2.2 Machine learning based approach 7

2.3 Hybrid approach 8

2.4 Related Works and Identification of research gap 8

Chapter 3: Design & Methodology **11**

3.1 Methodology 11

3.2 Proposing model/design 12
3.3 Creating Data set 13
3.3.1 Data Collection 13
3.3.2 Manually Annotations of Dataset 14
3.4 Hate Speech Detection 15
3.4.1 Preprocessing 15
3.4.2 Feature Extraction 15
3.4.3 Classifiers 16
3.4.4 Evaluation 17
Chapter 4: Implementation 19
4.1 Preprocessing 19
4.1.1 Open the csv File 20
4.1.2 Remove the HTML tags 20
4.1.3 Remove Non alphabetic Characters 21
4.1.4 Tokenizing 21
4.1.5 Remove Special Characters 21
4.1.6 Remove Stop Words 22
4.1.7 Stemming 23
4.2 Feature extraction 23
4.2.1 CountVectorizer – Bag of Word Features (BoW) 23
4.2.3 Tf-idf Vectorizer – Term Frequency Features (Tf-idf) 24
4.3 Classification Models and Evaluation 24
Chapter 5 - Results and Evaluation 25
5.1 countvectorizer 26
5.1.1 Logistic Regression 26
5.1.2 Multinomial Naive Bayes Classifier 27
5.1.3 Linear SVM 28
5.1.4 Random Forest Classifier 29
5.2 Tf-idf Vectorizer 30
5.2.1 Logistic Regression 30

5.2.2 Multinomial Naive Bayes Classifier 31
5.2.3 Linear SVM 32
5.2.4 Random Forest Classifier 33
5.3 Evaluating Classifier methods with Difference N-gram values 34
5.3.1 countvectorizer 34
5.3.2 Tf-idf Vectorizer 35
5.4 Summary of results 37
5.4.1 Summary Of countvectorizer 37
5.4.2 Summary Of Tf-idf Vectorizer 38
5.4.3 Summary 39
Chapter 6: Conclusion and Future Works 40
6.1. Conclusion 40
6.2. Future work 40
References 41

List of Figures

Figure 3.1.2: Design of the Research

Figure 3.3.1.1: Data Set Distribution According to the Language

Figure 3.3.2.1: Data Set Categorizing As the Hate or Not

Figure 3.4.1.1: Preprocessing Steps used in this Research

Figure 4.1.1: Code 02- Convert the Sinhala Sentence to Romanized Sinhala (Singlish)

Figure 4.1.3: Code 04- remove the Non alphabetic characters

Figure 4.1.4: Code 05- Tokenized Text

Figure 4.1.5: Code 06- Remove Special Characters

Figure 4.1.6: Code 07- Remove Stop words

Figure 5.0 Structure of the training and testing dataset.

Figure 5.1 Structure of the training Dataset

List of Tables

Table 3.3.2: Data Set Categorizing According to the Hate or Not

Table 5.1.1.0: Result of countvectorizer Logistic Regression classifier

Table 5.1.1.1: Confusion matrix of countvectorizer Logistic Regression classifier

Table 5.1.1.2: Classification report of countvectorizer Logistic Regression classifier

Table 5.1.1.3: Train and Test Accuracy of countvectorizer Logistic Regression classifier

Table 5.1.2.0: Result of countvectorizer Multinomial Naive Bayes Classifier

Table 5.1.2.1: Confusion matrix of countvectorizer Multinomial Naive Bayes Classifier

Table 5.1.2.2: Classification report of countvectorizer Multinomial Naive Bayes Classifier

Table 5.1.2.3: Train and Test Accuracy of countvectorizer Multinomial Naive Bayes Classifier

Table 5.1.3.0: Result of countvectorizer Linear SVM Classifier

Table 5.1.3.1 Confusion matrix of countvectorizer Linear SVM Classifier

Table 5.1.3.2 Classification report of countvectorizer Linear SVM Classifier

Table 5.1.3.3 Train and Test of countvectorizer Linear SVM Classifier

Table 5.1.4.0: Result of countvectorizer Random Forest Classifier

Table 5.1.4.1: Confusion matrix of countvectorizer Random Forest Classifier

Table 5.1.4.2: Classification report of countvectorizer Random Forest Classifier

Table 5.1.4.3 Train and Test Accuracy of countvectorizer Random Forest Classifier

Table 5.2.1.0: Result of Tf-idf Vectorizer Logistic Regression classifier

Table 5.2.1.1: Confusion matrix of Tf-idf Vectorizer Logistic Regression classifier

Table 5.2.1.2: Classification report of Tf-idf Vectorizer Logistic Regression classifier

Table 5.2.1.3: Train and Test Accuracy of Tf-idf Vectorizer Logistic Regression classifier

Table 5.2.2.0: Result of Tf-idf Vectorizer Multinomial Naive Bayes Classifier

Table 5.2.2.1: Confusion matrix of Tf-idf Vectorizer Multinomial Naive Bayes Classifier

Table 5.2.2.2: Classification report of Tf-idf Vectorizer Multinomial Naive Bayes Classifier

Table 5.2.2.3: Train and Test Accuracy of Tf-idf Vectorizer Multinomial Naive Bayes Classifier

Table 5.2.3.0: Result of Tf-idf Vectorizer Linear SVM Classifier

Table 5.2.3.1 Confusion matrix of Tf-idf Vectorizer Linear SVM Classifier

Table 5.2.3.2 Classification report of Tf-idf Vectorizer Linear SVM Classifier

Table 5.2.3.3 Train and Test of Tf-idf Vectorizer Linear SVM Classifier

Table 5.2.4.0: Result of Tf-idf Vectorizer Random Forest Classifier

Table 5.2.4.1: Confusion matrix of Tf-idf Vectorizer Random Forest Classifier

Table 5.2.4.2: Classification report of Tf-idf Vectorizer Random Forest Classifier

Table 5.2.4.3 Train and Test Accuracy of Tf-idf Vectorizer Random Forest Classifier

Table 5.3.1.13: Result of countvectorizer unigram

Table 5.3.1.14: Result of countvectorizer Train and Test Accuracy unigram

Table 5.3.1.15:Result of countvectorizer bigram

Table 5.3.1.16: Result of countvectorizer Train and Test Accuracy bigram

Table 5.3.1.17: Result of countvectorizer trigram

Table 5.3.1.18: Result of countvectorizer Train and Test Accuracy trigram

Table 5.3.1.18: Result of countvectorizer Train and Test Accuracy trigram

Table 5.3.2.14: Summary Result of Train and Test Accuracy unigram

Table 5.3.2.15: Result of Tf-idf Vectorizer bigram

Table 5.3.2.16: Result of Tf-idf Vectorizer Train and Test Accuracy bigram

Table 5.3.2.17: Result of Tf-idf Vectorizer trigram

Table 5.3.2.18: Result of Tf-idf Vectorizer Train and Test Accuracy trigram

Table 5.4.1.3: Summary Result of countvectorizer Logistic Regression

Table 5.4.1.6: Summary Result of countvectorizer Multinomial Naive Bayes Classifier

Table 5.4.1.9: Summary Result of countvectorizer Linear SVM Classifier

Table 5.4.1.12: Summary Result of countvectorizer Random Forest Classifier

Table 5.4.1.13: Final Result of Countvactorizer

Table 5.4.1.3: Summary Result of Tf-idf Vectorizer Logistic Regression

Table 5.4.1.6: Summary Result of Tf-idf Vectorizer Multinomial Naive Bayes Classifier

Table 5.4.1.9: Summary Result of Tf-idf Vectorizer Linear SVM Classifier

Table 5.4.1.12: Summary Result of Tf-idf Vectorizer Random Forest Classifier

Table 5.4.1.13: Final Result of Tf-idf Vectorizer

Table 5.4.3: Final Result

List of Acronyms

NLP	Natural Language Processing
NLTK	Natural Language Processing Tool Kit
SVM	Support Vector Machine
MT	Machine Translation
SA	Sentiment Analysis
MT	Machine Translation
BOW	bag-of-words
FB	Facebook
TN	True Negatives
TP	True Positives
FP	False Positives
FN	False Negatives
H-SWN	Hindi-SentiWordNet
DF	Data Frame
CSV	Comma Separated Values File
HTML	Hyper Text Markup Language
Tf-idf	Term Frequency Features
Tf	Term Frequency

Chapter 1 – Introduction

1.1 Background to the Research

1.1.1 What is social media?

Social media can be determined by the interaction and communication of content created by users. In modern society use of social networks has become essential in everyday activity. It ordinarily used to gain access to social interactions, news, and information to help make decisions. It the value of a communication tool with global stakeholders, sharing, creating and dissemination of information. Essentially, social media greatly affects our ability to build relationships access to information, dissemination and access to the possible decisions.

Also it is computerized technologies that facilitates the creation and sharing of information, ideas, and professional needs and other publications through virtual communities and networks. Social media is a virtual life for people and a place where people can express their feelings, opinions and beliefs. Examples of different types of social media are web pages devoted to forums, microblogs, social networks and wikis. Examples of social media organization such as Facebook, YouTube, Twitter …. etc.

Social media has many differences compared to traditional electronic media such as paper-based or television broadcasting. Compared to quality, frequency, usability, reach, and firmness we can see differences. Social media is a conversational transmission systems, while other traditional media functions as monopoly transmission systems. Facebook, Google+, Viber, WeChat, Weibo, WhatsApp, YouTube, Reddit, Tumbler, Twitter, LinkedIn, Instagram are the most popular social media web sites with high demands from the registered users.

1.1.2 Social media and hate Speech

Many social media analysts and observers have pointed out a range of positive and negative impacts of social media. It helps to get connected with individuals and thousands of real online communities and engage in effective communication. It can be used as a tool for marketing and advertising for corporations, non-profit organizations, entrepreneurs, advocacy groups and political parties. In meantime social media can be used for online harassment, trolling and cyber bullying which lays the path for depression of individuals and groups which can be emphasized as negative impacts of social media.

It is really difficult to find a single internationally recognized meaning for hate speech. Hate speech is highly co-related with the freedom of expression of individual, groups, minority rights and concepts of dignity, liberty and equality. According to it, unable to exactly identify hate speech. According to the hate speech definitions, hate speech is statements that cause harm, discrimination, hostility, and violence based on an identified social group or demographic group. In some cases it is mentioned that Hate speech is a communication that insults people based on their membership in a particular group. Hate speech can include images, videos, songs, stories. Hate speech attacks based on race and religion, sexual orientation or gender.

According to many rules and legislations in many countries hate speech is illegal. But it depends on the definition of hate speech given by that particular country. Hate speech is rampant on the Internet, especially on social media. Meaning and content of hate speech is very similar in both online social media and real society. But online social media hate speech renders current laws and gender regulations in an ineffective manner in many cases when it is compared with offline media. As a positive impact of social media, we can consider social media as an asset in terms of democratic, dialogic expressions. But it can be used by extremist groups as an advantage for them to disseminate hateful content. However the impact of online hate speech are more intense than offline hate speech.

Social media for example Facebook, YouTube, Twitter has various principles to handle hate speech.

In the YouTube policies [1] they encourage free to comments any video and do not allow hate comments and make unpopular comments and Hate speech as a content that promotes hatred towards individuals or groups based on many characteristics. According to their guidelines they define the some content of hate speech and not hate speech. As the example, in YouTube is generally good at criticizing a nation-state, but hateful comments about people based solely on their ethnicity are not good. YouTube has given users few options to report about content which we feel that violate their policy which is define for the hate speech. We can flag the particular video or we can file an abuse report on particular content.

In Twitter policies [2] they have mentioned that they strictly prohibit the hate content. They also consider content including attributes like race, religion, disability, gender, sexual orientation, age, veteran status in a violence promoting manner as hate speech. Some of uses are publishing some post with hate content or villain content or terrorist related content, it can cause a strong negative action or cause harmful, obscene, or obscene content are subjected to their policies. Reviews of products, services, companies, or brands that focus on hate reaction, sensitive speech, or violence but do not advocate potentially negative commentary are not subjected to their policies.

Facebook [3] also consider content including race, gender, age, veteran status, sexual, religion, disability status in hatred promoting manner as hate. Facebook also has given few options to report any policy abuse. We can send a message who are responsible for the posting. We can un-friend the person to remove them. We can block some Account from our account. We can also report the some account if their Behavior abuse or use privacy settings

According to all the mentioned policies and regulations by different social media organizations, it is clear that there is significant need of removing hate content from the social media sites.

1.2 Definition for Hate Speech

The insights derived from the literature developed the definition of hate speech is , "*Hate speech is the usage of language to insult or spread hatred towards a particular group or individual based on religion, race, gender or social status*."

1.3 Problem Statement

The aim of this research is to develop an integrated 'checkpoint' module for social media, targeting the Sri Lankan community. The solution was carried out in two cases. As the First case is selection and training a classifier to analyze social media post into a pre-known number of classes. The second case is locating a collection of unclassified social media post in classified folders according to their content.

Unfortunately, hate speech is not new topic to the world. Social media and most of online another software or websites which are used to communication, that are play greater role of hatred crime. As the example for this hatred crime, a social media suspected, history of terrorist hate messages suggests that online communication is contributing the radicalization of their works [3]. Social media can play an even more direct role in some cases as the example, Video of suspected 2019 terrorist attack in the New Zealand Christchurch and he posted the live video that situation on Facebook [3]

A wide range of online communication platforms, with social media, allows them to post freely their mind, sometimes in a way that prevents a person from being identified by their real details. The ability to express oneself is a human right, inducing and spreading hatred toward another group is an abuse of this freedom. As the example, U.S. Bar Association argues, hostility is legal

and is protected by the First Amendment, although not if it directly requires violence [4]. As such, many online forums, such as Facebook, YouTube, and Twitter, find hate speech harmful and have a code of conduct to remove hate speech content [5] - [7]. Because of societal concerns and how hate speech is becoming more popular on the Internet is a powerful motivation to investigate the automatic detection of hate speech. By automating its detection, you can reduce the spread of hostile content.

1.4 Research Problem

There are several social media websites with thousands of registered users. Social media is an important part of the society, which connects many communities together. Through it supports the world to connect with each other, it has several negative effects on society as well. Hate speech has become one of the major issues in the social media. Considering the policies and rules that have been established by these social media organizations, it is said that there is a great need to automatically identify hate speeches, so that it can be benefited to identify the effect of hate speeches in various communities. It will assist to reduces the impact of hate speeches on different communities and individuals and allow people to engage in more online discussions without any fear or depression. Meanwhile, it will reduce the spread of bad feelings.

Deliberating one of the negative impact on social media to the community and the importance of reducing that effect, following research question have been identified to answer this problems.

- Is it possible to identify hate speech in social media automatically?
- How to use the lexicon based approach to correctly identification of hate speech?
- How to use a machine learning approach for hate speech identification?

Most of social media platforms have created user rules that prohibit hate speech. Following these rules requires a lot of manual work to review each report. Some platforms, such as Facebook, have recently increased the number of content moderators. Automated tools and approaches can speed up the review process or place human resources on positions that require close human scrutiny. In this section, we consider automatic approaches to detecting hate speech from text.

1.5 Motivation

The Organizations do not have a proper, effective and real-time methodology of detecting hate comments, post of the social media automatically. It is very easy for all if this gap was filled. Taking the initial step of reducing this gap by deciphering the customers and group them based on their choice is the main inspiration.

1.6 Research Contribution

1.6.1 Goal

According to the Universal Declaration, freedom of expression is regarded as a human right. of Human Rights. It is a basic pillar of every democratic society. Nevertheless the existence of hate speech in public deliberations is a direct indication of a democratically weak society. Hence,

there is a great need to identify online hate speech which can be paved the way for reducing cybercrime and the spread of hate in society. The mail goal of this project is to overcome the problem of Use machine learning techniques for the hate speech detection on social media. Collect reader responses of Sri Lankan articles on web.

1.6.2 Objectives

The main objectives of this study is to identify hate crimes remarks on Sri Lankan social media using an accurate and effective natural language processing (NLP) model. The objective are as follows:

- Manually annotate responses as hate speech or not.
- Classification of hate speech on social media.
- Identification of different types of hate speech detection.
- Design, implement, and evaluate a technological methodology for detecting Sri Lankan hate speech.

1.7 Scope and Limitations

The scope of this project is to provide a prototype to identify social media reports by categorizing hate speech. Hence the hate speeches are context dependent and language dependent, the expected inputs to machine learning algorithms are the size and content of the responses which contains Romanized Sinhala comments annotated with two labels:

(1) Hate speech
(2) Not a Hate Speech

For this, Sri Lankan websites have been selected and as the existing Sinhala reader responses are not enough, number of Sri Lankan articles from the Reader Response Collection have been collected as well. One such website is Colombo Telegraph website where considerable number of users have already registered. After collecting required data, used the machine learning techniques to combine word classification with a lexicon-based approach. By analyzing the inputs, the system would predict the success of the inputs to assist the decision of identifying hate speeches.

1.8 Justification for the Research

It is clear that number of social media websites get increased day by day. Meantime number of registered users gets increased day by day and the amount of internet-generated content is growing rapidly. It is really difficult to do manual flagging to remove hateful content in online media. So Accurate, automated methods must be used for the offensive flag hate speech in online media. When looking at the policies and regulations established by different social media websites also we feel that there's a big need in identifying hate speech automatically. Automatic identification of online hate will led the individuals to engage in more online discussions without any fear and depression while minimizing the impact on different communities as well as individuals. At the same time it will be helpful to decrease the spreading of bad feelings like terrorism and to reduce hate crime

1.9 Structure of the Dissertation

The chapters of this report include diagrams and diagrammatic descriptions to provide an overview of the project. After a comprehensive and detailed description and understanding of the research field and scope in Chapter 1.

In Chapter 2 include the detailed discussion of the study of literature in this field, which has been mentioned in the research process. The research described in this chapter is the current knowledge and new methods related to the research.

In chapter 3 we explain the design of the project which planned experimental set up and design is described include in this chapter. In here, we will explain how to create a data set and how to take measures for feature extraction.

In chapter 4 of this document followed by the implementation details.
In the chapter 5 detailed description about the results and evaluation criteria is presented in the project.

In the final chapter, the dissertation concludes by discussing the study's concluding remarks and future works.

Chapter 2: Literature review

This chapter provides the description about previously used methods for hate speech detection by using the computerized methods. When taking into consideration the shared through social media in the past few years, it is evident that a considerable percentage is belonging to the 'hate-speech' category. Social media analysts and observers have pointed out a range of positive and negative impacts of social media. Most of the previous researchers use many techniques to detect hate speech automatically. Although difficult to compare directly between different methodologies used in different studies.

In this Literature Review explain how data sets used, preprocessing protocols used and experimental setups built During the past recent years, there have been many types of research done on the automatically detect hate speech on social media.

2.1 Lexical based approach

This approach is the text classification and it is an important part of the task be able to identify the lexical phases. The machine is powered by models of language and grammar, rules created manually describes some types of texts or basic knowledge of the domain describes some types of texts. Vocabulary plays a major role on grammar in this approach.

For domains like sentiment analysis, there are inbuilt lexicons which are widely used. Those lexicons are comprised of different words and the polarity rates which indicate whether that word gives a feeling of negative or positive. Since, hate speech detection is a currently emerging research area in past few years still there are no such lexicons built to the detection of hate speech. There are only collections of words which are banned or recognized as hate words. But there are no rates given for the words indicating how much hate is expressed through that word. Google bad word list is the most widely used hate lexicon which is built by Google collecting the banned words by Google.

2.2 Machine learning based approach

Field of computer science which includes the topics of the computer's that can be learn without explicitly programmed is known as machine learning. In machine learning algorithms instead of programmer defining rules for particular task, data is fed to the machine and algorithm is adjusted in order to perform the task. So, machine learning is basically a data driven approach. Currently machine learning techniques are used in the many areas of computer science Also machine learning is the main part of the text mining.
Supervised learning and unsupervised learning are two main strategies of machine learning. When the input data is labeled it is called supervised learning and when input data is given without the label, it is called unsupervised learning. Supervised learning algorithms try to fit its

inner machinery to match the mapping function of the labeled data. The data set is divided to the two data set as training dataset and testing dataset. Algorithm tries to make predictions on training data until a considerable level of performance is achieved. This is known as learning phase and then it's going to be the testing phase. What happens in testing phase is the creation of predictions on the testing set and calculating the performance evaluation matrices to compare the predicted label and actual label.
Support vector machine, Naïve Bayes classifiers, Decision tree classifiers, Logistic regression models are few examples of supervised machine learning algorithms while kmeans clustering, self-organizing maps are grouped as unsupervised learning algorithms. Support vector machine (SVM) is the widely used supervised algorithm for the task of hate speech detection. Meantime hate speech detection has been framed as an supervised learning task since the number of researchers who have tried out unsupervised learning for hate speech detection is relatively very law.

2.3 Hybrid approach

Many researchers use hybrid approaches to detect hate speech. The combination of learning-based and lexical approaches is done here. In some cases, the first step is to use the "***lexical based approach***" used and filter the data then insert the filtered data into the machine learning model. Lexical resources can be apply to extract features from textual dataset and provided with machine learning format.

2.4 Related Works and Identification of research gap

Lexical approaches are the Text classification is an important part of the task be able to identify lexical phases. The machine is powered by models of language, grammar, rules genarated manually describes some types of texts or basic knowledge of the domain describes some types of texts. Vocabulary plays a major role on grammar in this approach.

One of the research done by using the lexical based approach[8] and create a model by using the sentiment analysis with subjectivity detection to detect the subjectivity of a sentence and polarity of the sentiment expressions and after to identify the hate, they used a lexicon build. By using different sources they collect the data such as 100 blog postings from 10 different websites are collected as the main source, selected websites from Hate Directory which is composed of sites to be generally offensive and other corpus is created using 150 page document websites.

The approach proposed in this research [4] consists of several approaches. As the one of the approaches subjectivity detection. A rule based and learning based approach is used in the subjectivity classifier. Sentiment lexicon resources of Wilson [9] and SentiWordNet [10] are used for this purpose. A list of over 800 subjective clues with several tags as the example positively, negatively and neutral are included into that sentiment lexicon. If a sentence contains two or more strongly subjective clues, the sentence is classified as a strong subjective clue. Beside this, They give a few points for the words in the sentence as negative and positive and To get the synchronization score, subtract the negative from the positive. Also as the next step of the

proposed approach a lexicon for hate speech is built. Here they built a rule based hate speech classifier which relies on three different sets of features. It negatives polarity words, hate verbs and theme based grammatical patterns to create negative polarity features they have used subjective sentences with negative semantic orientation identified in first step and extracted all verbs which have a relation with hate verbs in their hate corpora to prepare the feature set hate verbs. But by using machine learning approaches such as SVM maximum entropy can be applied directly with a possibility of increasing precision and recall scores. Also when creating the lexical tags they used subjectively clues and it is best practice for text mining but here not methods for hate full symbols detection approaches.

The development of a sentiment lexicon was done by Joshi [11] for the Hindi language, they proposed the fallback strategy to perform sentiment analysis on Hindi language files and used three methods for sentiment analysis such as

- In-language Sentiment Analysis (SA)
  This approach tells you how SA works if the classification training is conducted in the same language with the text corpus (Hindi classification format using a training corps in Hindi).
- Machine Translation (MT)
  By using this approach used the translation module to transfer the Hindi documents to English documents. In this approach they used the six steps, such as Training corpus, Model, Manually Annotated corpus, transfer files Hindi to English, Translate Hindi corpus, Polarity Detection. In here train the classifier for the English based files.

- Resource-based sentiment analysis.
  "develop a lexical resource called Hindi-SentiWordNet (H-SWN) and implement a majority score based strategy to classify the given document.it is evident that machine learning-based approaches are better suited for sentiment prediction compared to resource-based approaches . All these approaches need a large amount of training data. A good resource-based classifier can act as a substitute for this large amount of data".

Most of researchers used mainly consider three approaches and provide superior classification performance compared to majority systems based on lexical resources and constitute the fallback strategy for SA in Hindi. By using this in- language process Sentiment Analysis can apply for the Singlish language.
Another research is conducted by Z. Waseem[12] to propose a logistic regression classifier and cross-validation were used to test the effect of different components on predictive performance, and their research found the character n-gram was better than the word n-gram. They used gender, location and length as the main features of Twitter. Extra features are available with the best performance up to 4 grams with gender. The use of additional features did not improve location and length F1 scores. What they have concluded is their solution can be useful in some cases but not for all and the problem can be partially solved using a character n-gram based approach.

In 2018 done Sinhala hate speech detection research [13] they done this research by using the Sinhala Unicode and automatically detect the hate speech in Sinhala comments. As the

classification model they used the SVM and dataset trained using the random selected data and other data used as a test data set. In detecting hate speech, it is not enough to identify speech based on racism, because hate speech define by many characteristics.

The basic method to detecting hate speech is use a keyword-based method. Using ontologies or dictionaries, text containing potential hate keywords can be identified. As the example the hate base [14] maintains a database of malicious terminology covering most groups of languages. As the terminology changes over time, well-maintained resources become invaluable. However, as we have observed in studying the detection of identifying the hate speech, hate speech alone isn't sufficient to produce hate speech. Keyword-based approaches that is quick and easy to identify. Only racial slurs detection would be accurate with low recall values.

Most Of researchers widely used the machine learning and lexicon based models for the text mining Also only few of researchers studying on deep learning approached for the text mining [15]. Supervised learning is used in many experiments with studying about the lexicon based approach and there is still insufficient research on unsupervised learning to identify hate speech.

A thorough search of the Internet revealed that most of the research was conducted to identify hate speech systems for social networks in different countries for their native languages.
Despite these findings, it is clear that Sri Lanka hasn't relatively detection systems to detect Romanized Sinhala hate speech. In this regard, the purpose of this study is to investigate the development of a detection system to reduce this research gap and identify Romanized Sinhala hate speech. The results of this study allow the user to determine the extent to which the user may indirectly hate behavior, thereby minimizing the negative psychological impact on another community or another.

Chapter 3: Design & Methodology

3.1 Methodology

In this project will look at difference ways to identify the hate related speech on many social media comments and distinguish it from ordinary pornography. For this purpose, aim is to create lexicon bases by using the data set which from the gathered resources such as Facebook, tweeter, etc. In Figure 3.1.1 represents the *Structure of the Methodology and* through the approach the supervised classification method. Then those data set manually labeled as two categories, hate speech or not. The dataset for use during the training phase and the test phase. The dependency of result of training phase, it moves to the testing phase for the classification.

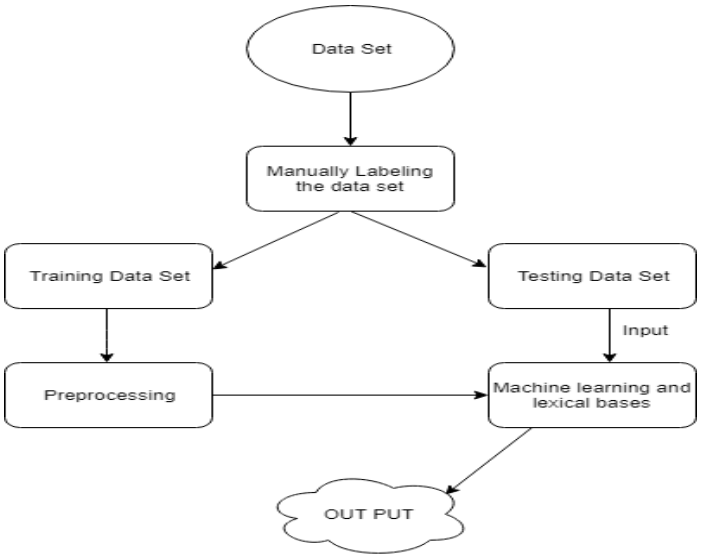

Figure 3.1.1: Structure of the Methodology

Preprocessing of the data set is a pre requisite of the project. For that purpose, eliminating from, unstructured data which contains typos, non-standard acronyms and mutual meanings will consider in this preprocessed stage. After that use the feature extraction method to extract desired information from the data set and it will be an important role in this process.

3.2 Proposing model/design

In this part description will be given on dataset used in experiment explaining origin of data and annotation process of data. Then data preprocessing and feature extraction steps will be presented in detail. The entire details about the design of the experimental setup, algorithms used will be presented.

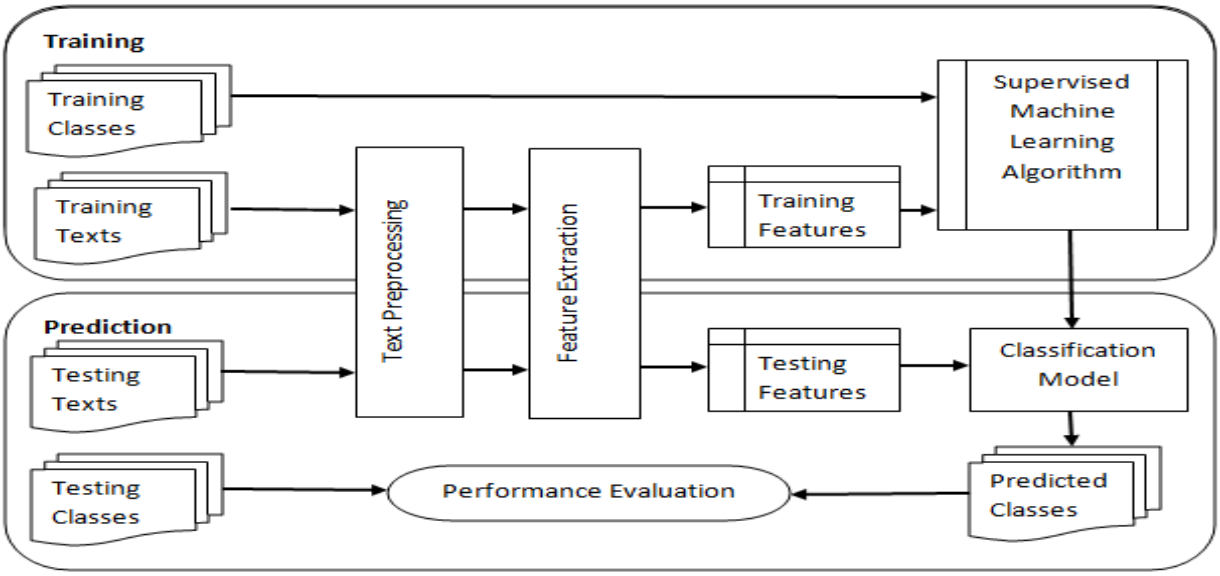

Figure 3.1.2: Design of the Research

Figure *3.1.2* presents the high level overview of the system design for the model built using an unsupervised machine learning algorithm. Here the text preprocessing and feature extraction steps will be same as the steps in supervised learning model. Only the way learning is changed.

3.3 Creating Data set

3.3.1 Data Collection

To perform a successful experiment on hate speech detection availability of a labeled corpus is really important. Collected data set is different articles consist of comments written by users in social media as the example YouTube ,Facebook and Twitter, Etc. that is articles based on Sri Lankan matters also all comments are written in Sinhala or Romanized Sinhala language. Here I collect both category because of most of Romanized Sinhala comments includes the Sinhala words and If they insert the Sinhala words then I Converts those Sinhala words to Romanized format by using the unidecode .

Category	**Number Of Comments**
Sinhala Comments	0875
Romanized Sinhala Comments	1000
Both Sinhala and Romanized Sinhala Insert in the comments	0125
Total Number Of Comments	2500

Table 3.3.1: Data Set Distribution According to the Language

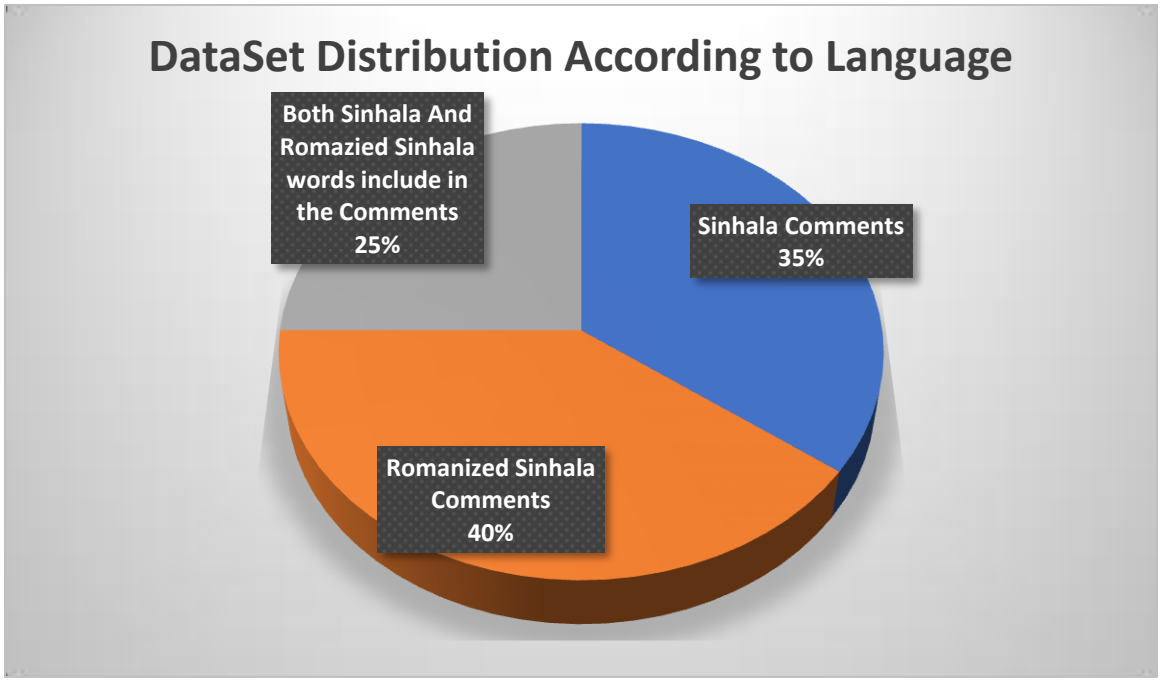

Figure 3.3.1.1: Data Set Distribution According to the Language

3.3.2 Manually Annotations of Dataset

Each data in the data set will determine if there is a hate story. There are only two tags that hate or not. The Collected data that contains hate speech will be identified as If that comment is hate speech labeled as the “Yes” and if it is not hate speech labeled as the “No”. The dataset was annotated manually. The Collected dataset consist of 2500 data. Among them, 1,400 data have been manually annotated as hate speech and 1100 comments annotated as the hate speech.

Category	**Number Of Comments**
Hate Speech Comments	1400
Not Hate Speech Comments	1100
Total Number Of Comments	2500

Table 3.3.2: Data Set Categorizing According to the Hate or Not

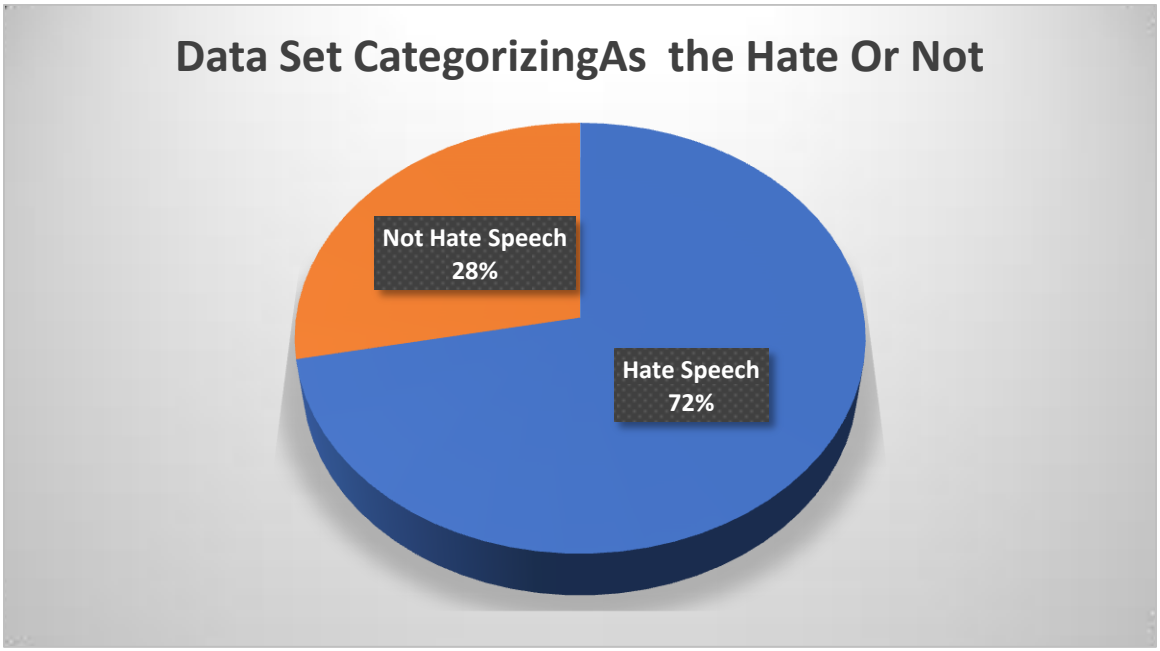

Figure 3.3.2.1: Data Set Categorizing As the Hate or Not

3.4 Hate Speech Detection

3.4.1 Preprocessing

Figure 3.4.1.1: Preprocessing Steps used in this Research

To reduce noise, all text data should be cleaned before they are apply to the classifiers. Preprocessing is one of the main step in the text processing. BeautifulSoup is the one of the best preprocessing libraries which can be used for this task. By using the BeautifulSoup Remove the HTML tags, remove non alphanumeric characters , Convert the All text data to Romanized Sinhala, Remove Special Characters, Tokenization, stops word removal, Stemming are basic preprocessing steps which were used in this research.

3.4.2 Feature Extraction

We have explored the n-gram feature types in our experiment. In the feature extraction, create bag-of-words(BOW) representations of the data Set In here utilize the word n-gram Also implement for the n=1 (word unigram), n=2 (word bigram) and n=3 (word trigram) and predict and evaluate model using tokenized data set. Also calculate the accuracy, precision, recall and f1 measure, and confusion matrix and classification report for the data set.

3.4.3 Classifiers

We used the supervise learning technique for the detecting hate Speech in Romanized Sinhala language. Also compare the performance of following Algorithms.

- Support Vector Machine[16]
  Support vector machine (SVM) and logistic regression are linear classifications that can predict categories based on a set of scores for each component. There are open source implementations of some models, such as the models in the famous Python machine learning package *sklearn toolkit*.
  The Support Vector Machine we use in our study is LinearSVC. SVM is best suited when measurements exceed the number of samples. Making a single linear plane in the x-dimensional space where each x features of a given feature set corresponds to one dimension in an x-dimensional space is the main task accomplished by SVM. The plane should be positioned in way such that very few numbers of samples are on wrong side of the plane.

- Logistic Regression[17]
  "The logistic model is used to model the probability of a certain class or event existing such as Yes or No Also Logistic Regression are linear classifiers that predict classes based on a combination of scores for each feature". Logistic regression is used to when the target is categorical such as in this research our target is predict the given input is hate or not. So I used the Logistic Regression method for the experimental.

- Random Forest Classifier[18]
  This is the supervised learning algorithm and Random Forest Classifier used to classification, regression and feature selection. In this Algorithm it select the random samples from the given data set and create the decision tree for the random created data samples, then calculate a prediction result from every decision tree. This is considered as a highly accurate and eliminate the overfitting problems. So I selected this as the classification method for this experiment.

- Naïve Bayes Classifier[19]
  Naïve Bayes models probability under the assumption that probabilities do not affect each other.
  Naïve Bayes Classifiers are based on Baye's theorem with "naïve" assumption and it is a supervised learning algorithm. Naïve assumption is all features are independent of each other. We used both Gaussian Naïve Bayes and Multinomial Naïve Bayes for the experiment.

3.4.4 Evaluation

The Natural Language Processing (NLP) community is making extensive use of the resources on the internet. As NLP research gets the attention of the general public, we need to make sure that our results are solid and reliable [20].

The important question is what happened to the data and how reliable it was from the data. Therefore, we conducted a quantitative analysis of the frequency of data collection, the method of data publish and the type of data[21].

We follows a quantitative approach so this project scope is less in-depth data across a larger number of study participants and Collect data using structured instruments, what we do is a systematic investigation where we can use statistical, mathematical techniques to accomplish our task. Collected dataset is annotated manually and then fed to the built model in order to get predictions. So, then we had to analyze the data and results with an evaluation metric in order to check the performance of the model, biasness of the results and to what extent we can generalize our results [22].

The evaluation metric we established were used throughout the experiment. This study is related to the accuracy of data analysis and natural language processing (NLP), the accuracy, recall rate and F score were selected as evaluation metric. Our data set, comments contain or do not contain hated.

All values that we test for accuracy, precision, recall, and F-scores depend on the concept of positive and negative. We define positive as a hate speech and negative as a does not a hate.

		Predicted Class	
		With Hate	No Hate
True Class	Hate Speech	True positive	False Positive
	Not Hate Speech	False Negative	True Negative

Table 3.4.4.1: Structure of Confusion Matrix

True Negatives (TN), True Positives (TP), False Negatives (FN) and False Positives (FP) are defined as given in the Table 3.4.4.1. At the same time we observed the confusion matrix which was built according to the above table.

Accuracy = (TP + TN) / (TP + TN + FP + FN)

According to the given formula accuracy can be defined as the fraction of predictions that are correct.

Although accuracy is used in many natural language processing researches for evaluation, there are few problems with accuracy which are very common. The main problem is accuracy is not a good measure of the classes of data is unbalanced.

56% of our annotated data is in no hate class while 44% of annotated data belong to hate class. The data set is not 100% balanced, but according to the percentages of data it is fairly balanced.

Category	Number Of Comments	Percentages Of Data
Hate Speech Comments	1400	56%
Not Hate Speech Comments	1100	44%

Table 3.4.4.2: percentages of data Balance

So, we can use accuracy as a measure. Accuracy measure gives more weight to the correctly classified positives and negatives, so that when the data set is unbalanced it can give a higher accuracy considering one class although other class is misclassified.

Precision (P) = TP / (TP + FP)

Fraction of predicted hate comments which were actually hate comments is defined as precision. From this measure we can observe how correct the positive predictions are. To look over the correctly predicted positives precision is the best measure to use, since it does not consider about negatives.

Recall(R) = TP / (TP + FN)

The fraction of hate comments that were detected is known as recall. From this measure we can get the idea of number of hate comments identified and number of hate comments the classifier missed. Since recall also does not consider about true negatives this measure is also better for our task.

F-score = (2 x R x P) / (R + P)

When harmonic mean of precision and recall is calculated we called it as F-score. F-score ensures that we will not overly rely on either precision or recall. So, that we have considered F-score as our main evaluation measure.

Chapter 4: Implementation

In this chapter we have described the various components that go into constructing the model and making classifications and description of all the implementations, codes, processes used and technologies used to the prediction. Also Implementation task of this system can be divided into sub goals preprocessing and feature extraction, training classifier, evaluation. The main steps of the experiment are as follows.

- Data collecting and manually data annotation
- Convert All data to the Romanized Sinhala format
- Data preprocessing and convert All data to specific word format.
- Feature Extraction
- Build the classification model and evaluation

Details of the first step data collection and annotation were discussed in the Design chapter. Also in preprocessing steps are describe in design chapter in this implementation chapter include the detailed description of preprocessing and classification of the research. So this chapter will continue from preprocessing onwards.

The chosen language is Python (python 3) because of these are available more libraries which are most useful for Text processing Algorithms. BeautifulSoup offers most of the preprocessing activities which is very important in text analytics sklearn toolkit offers implementations of Support vector machine and feature extraction techniques like BoW.

4.1 Preprocessing

The proposed solution will be the development of a core Natural Language Processing (NLP) module that specifically detects Romanized Sinhala posts and comments as hate speech or not.

The first step of building up the above would be the accumulation of ‘hate-speech’ and ‘non-hate-speech’ content posted or shared by individual users through Facebook and other social media so far, into one data reserve Manually annotate the collected speech phrases as hate speech or not.

The dataset will then be subjected to a data pre-processing in order to remove all unnecessary words and symbols and perform tokenization of words using the text processing python library. Define the stop words and stems for the Romanized Sinhala language. Then by using the Python and BeautifulSoup library we can apply the pre-processing for the dataset. We stored the preprocessed data in-memory then it can be directly accessed for the next steps of the experiment.

4.1.1 Open the csv File

All the comments were stored in a csv file. So, before text preprocessing we had to read all the comments in the csv file into a dataframe in python. Then all the next activities were done using this dataframe. Read the csv file By using pandas library in python.

```
import pandas as pd

hateSpeechData = pd.read_csv("Hate_Speech.csv", index_col ="PhraseNo")
```

Code 01: Open csv File

Convert Sinhala data to Romanized Sinhala format

```
# this function use to convert the sinhala words to shinglish

def convertSinhalaToSinglishDf(inputDf):
    for index in range(len(inputDf)):
        currTockenList = inputDf['Phrase'].values[index]
        singlishTokenList = []
        for token in currTockenList:
            singlishTokenList.append(unidecode(token))

        inputDf['Phrase'].values[index] = singlishTokenList;
    return inputDf
```

Figure 4.1.1: Code 02- Convert the Sinhala Sentence to Romanized Sinhala (Singlish)

4.1.2 Remove the HTML tags

As the second step of preprocessing remove the HTML tags in the comments and return the dataframe after that and implementation given below

```
# this function use to remove html tags

def removeHtmlTags(inputDf):
    for i in range(len(inputDf)):
        currentPhase= inputDf['Phrase'].values[i]
        inputDf['Phrase'].values[i] = BeautifulSoup(currentPhase,"html.parser").get_text()

    return inputDf
```

Figure 4.1.2: Code 03- HTML tag removal

4.1.3 Remove Non alphabetic Characters

As the next step of preprocessing remove the Non alphabetic characters in the comments and return the dataframe after implementation given below.

```
# this function use to remove non-alphanumeric characters

def clearNonAlphanumericCharacters(inputDf):
    nonAlpha = ['~','!','"','#','$','%','&','\'','(',')','*','+',',','-','.','/',':',';','<','=','>','?','@','[','\\',']','^','_','`','{','|','}']

    for i in range(len(inputDf)):
        currentValue = inputDf['Phrase'].values[i]
        for ch in nonAlpha:
            if ch in currentValue:
                currentValue = currentValue.replace(ch,' ')
            #   print(type(currentValue))

        inputDf['Phrase'].values[i] = currentValue
    return inputDf
```

Figure 4.1.3: Code 04- remove the Non alphabetic characters

4.1.4 Tokenizing

Breaking up strings into words and punctuations is known as tokenization. We tokenize words in every comment. For this task use the below implementation,

```
# this function use to tokenized the phrases in dataframe

def tokenizedText(inputDf):
    for i in range(len(inputDf)):
        currentPhase= inputDf['Phrase'].values[i]
        tokenizedList = []
        for curr in currentPhase.split():
            tokenizedList.append(curr)
        inputDf['Phrase'].values[i] =  tokenizedList
    return inputDf
```

Figure 4.1.4: Code 05- Tokenized Text

4.1.5 Remove Special Characters

Special characters are removed by checking with Romanized writing styles. Most of peoples are write “A” and “E” characters in same sound Such as,

- “Atha” = “etha” =<Insert the Sinhala word here> So here I remove the “A” and “E” characters
- “Tha”= “Ta”=< Insert the Sinhala word here > So here I remove the “H” caharacter
- “ Sh”= “S”=< Insert the Sinhala word here > So here I remove the “H” caharacter
- “Ch”= “C”=< Insert the Sinhala word here > So here I remove the “H” caharacter

- “W”= “V” So I replace the “V” to “W”
- Remove the Numbers
- If single character occurd as the word then I Remove the Single characters from the data set.

```
# this function use to remove useless characters in the Romanize singlish.

def removeUselessCharacters(inputDf):
    for i in range(len(inputDf)):
        currentValue = inputDf['Phrase'].values[i]
        # print(currentValue,"==>") kickoff = [item.replace("'", "") for item in kickoff]
        # currentValue = currentValue.translate({ord('a'): None})
        currentValue = [c.replace('th','t') for c in currentValue]
        currentValue = [c.replace('ch','c') for c in currentValue]
        currentValue = [c.replace('a','') for c in currentValue]
        currentValue =[c.replace('e','')for c in currentValue]
        currentValue = [c.replace('v','w')for c in currentValue]
        inputDf['Phrase'].values[i] = currentValue
    return inputDf
```

Figure 4.1.5: Code 06- Remove Special Characters

4.1.6 Remove Stop Words

Stop words are removed by checking with the stop word csv file.

```
#remove stop words in the dataframe using given stop word array and return dataframe
def removeStopWords(inputDf,stopWords):

    for i in range(len(inputDf)):
        currTockenList = inputDf['Phrase'].values[i];
        filteredTockenList = [w for w in currTockenList if not w in stopWords]
        inputDf['Phrase'].values[i] = filteredTockenList
    return inputDf
```

Figure 4.1.6: Code 07- Remove Stop words

4.1.7 Stemming

"*Stemming is the process of reducing a word to its word stem that affixes to suffixes and prefixes or to the roots of words known as a lemma*."

```
#stem the input dataframe
def stemmingDf(inputDf):
    print("*** Stemming ***")
    stemmingWordDic = getStemmingWordDic()
    for i in range(len(inputDf)):
        currTockenList = inputDf['Phrase'].values[i];
        print("currTockenList  ",currTockenList)
        stemmedTockenList=[]
        for token in currTockenList:
            stemmedToken = stemmingWordDic.get(token,"")
            print("Token  ",token)
            if stemmedToken != "":
                stemmedTockenList.append(stemmedToken)
                print("stemmedToken  ",stemmedToken)
            else:
                stemmedTockenList.append(token)

        inputDf['Phrase'].values[i] = stemmedTockenList
    return inputDf
```

Figure 4.1.7: Code 08- Stemming

4.2 Feature extraction

By using the sklearn toolkit in python do the feature extraction activities. The feature extraction codes basically rely on the functions of sklearn toolkit as follow.

- CountVectorizer
- Tf-idf Vectorizer

4.2.1 CountVectorizer – Bag of Word Features (BoW)

CountVectorizer implements both tokenization and occurrence counting in a single class. Simply we can convert a collection of text documents to a matrix token counts using CountVectorizer. Using this vector space model we can get the idea of the unique words in all comments and the frequency of each term in vector. So, bag-of-words features were extracted using countvecorizer. Then we extracted unigram,bigram,trigram features also using this vectorizer.[23]

4.2.3 Tf-idf Vectorizer – Term Frequency Features (Tf-idf)

Term frequency-inverse document frequency vector is a way to measure the importance of a word or term. We can check how rare a word is present in a document using tfidf. So, using this vectorizer we can have words with highest importance as a feature. The specialty of Tfidf is frequency of the term is off-set by the frequency of the word in the corpus which clearly says that some words appear more frequently in general.[24]

```
from sklearn.feature_extraction.text import CountVectorizer,TfidfTransformer,TfidfVectorizer

from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from unidecode import unidecode
from sklearn.metrics import accuracy_score


from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import SGDClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
```

Figure 4.2.3: Code 09- functions of sklearn toolkit used for implementation

4.3 Classification Models and Evaluation

We used supervise learning technique for the detecting hate Speech in Romanized Sinhala language. Also here I compare the performance of the selected four Algorithm: Logistic Regression, Multinomial Naive Bayes Classifier, Linear SVM, and Random Forest Classifier using the collected dataset with unigram,bigram,trigram teachers and different min-Df values.

Chapter 5 - Results and Evaluation

In the result and evaluation section will be comparing different four classifiers model and different feature sets with regard to accuracy of the data set, precision of the data set, recall of the data set and F-score of the data set measures.
All the comments were stored in one csv file and both training and testing data were stored in a single csv file. After reading the csv file into a DF, data was divided into training data set and testing dataset using a function.

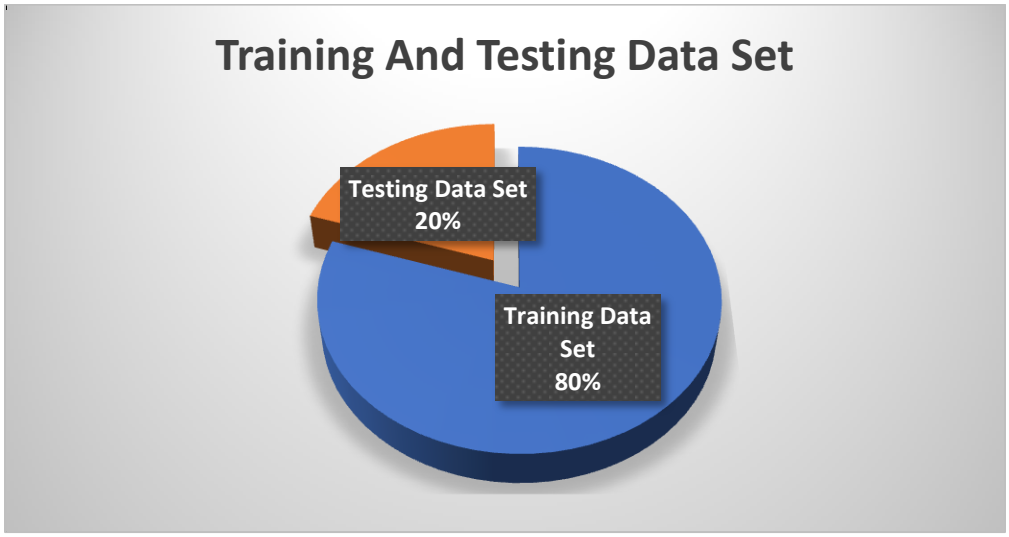

Figure 5.0 Structure of the training and testing dataset.

The prepared training dataset consist of 500 comments and among those training data set 350 comments are manually annotated as hate speech and 150 comments annotated as the not hate speech.

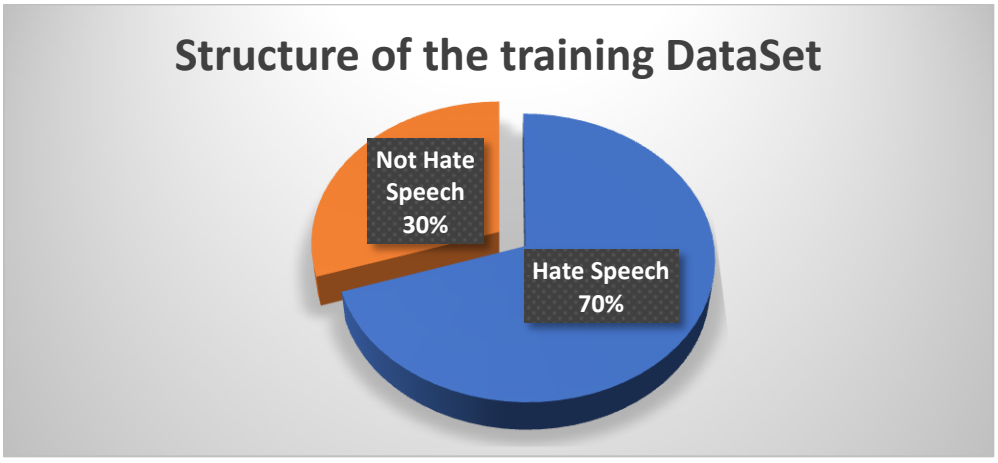

Figure 5.1 Structure of the training Dataset

All features is extracted using **countvectorizer and Tf-idf Vectorizer** in sklearn toolkit. Then same feature vector is passed for four different Algorithms and using testing data performance of the models was evaluated by using **deferent N-gram values and Min-DF values is 3**.Below results found from the unigram feature.

5.1 countvectorizer

5.1.1 Logistic Regression

Accuracy	0.65
Precision	0.65
Recall	0.65
F1 Score	0.64

Table 5.1.1.0: Result of countvectorizer Logistic Regression classifier

Confusion matrix

		Predicted Class	
True Class		Hate	Not Hate
	Hate	212	60
	Not Hate	115	113

Table 5.1.1.1: Confusion matrix of countvectorizer Logistic Regression classifier

Classification report

	precision	recall	f1-score	support
NO	0.65	0.78	0.71	272
YES	0.65	0.50	0.56	228

Table 5.1.1.2: Classification report of countvectorizer Logistic Regression classifier

Train Accuracy	0.993
Test Accuracy	0.65

Table 5.1.1.3: Train and Test Accuracy of countvectorizer Logistic Regression classifier

5.1.2 Multinomial Naive Bayes Classifier

Accuracy	0.73
Precision	0.73
Recall	0.73
F1 Score	0.73

Table 5.1.2.0: Result of countvectorizer Multinomial Naive Bayes Classifier

Confusion matrix

		Predicted Class	
True Class		Hate	Not Hate
	Hate	207	65
	Not Hate	68	160

Table 5.1.2.1: Confusion matrix of countvectorizer Multinomial Naive Bayes Classifier

Classification report

	precision	recall	f1-score	support
NO	0.75	0.76	0.76	272
YES	0.71	0.70	0.71	228

Table 5.1.2.2: Classification report of countvectorizer Multinomial Naive Bayes Classifier

Train Accuracy	0.969
Test Accuracy	0.734

Table 5.1.2.3: Train and Test Accuracy of countvectorizer Multinomial Naive Bayes Classifier

5.1.3 Linear SVM

Accuracy	0.68
Precision	0.68
Recall	0.68
F1 Score	0.67

Table 5.1.3.0: Result of countvectorizer Linear SVM Classifier

Confusion matrix

		Predicted Class	
		Hate	Not Hate
True Class	Hate	219	53
	Not Hate	109	119

Table 5.1.3.1 Confusion matrix of countvectorizer Linear SVM Classifier

Classification report

	precision	recall	f1-score	support
NO	0.67	0.81	0.73	272
YES	0.69	0.52	0.59	228

Table 5.1.3.2 Classification report of countvectorizer Linear SVM Classifier

Train Accuracy	0.9975
Test Accuracy	0.676

Table 5.1.3.3 Train and Test of countvectorizer Linear SVM Classifier

5.1.4 Random Forest Classifier

Accuracy	0.66
Precision	0.67
Recall	0.66
F1 Score	0.64

Table 5.1.4.0: Result of countvectorizer Random Forest Classifier

Confusion matrix

		Predicted Class	
		Hate	Not Hate
True Class	Hate	232	40
	Not Hate	129	99

Table 5.1.4.1: Confusion matrix of countvectorizer Random Forest Classifier

Classification report

	precision	recall	f1-score	support
NO	0.64	0.85	0.73	272
YES	0.71	0.43	0.54	228

Table 5.1.4.2: Classification report of countvectorizer Random Forest Classifier

Train Accuracy	1.0
Test Accuracy	0.662

Table 5.1.4.3 Train and Test Accuracy of countvectorizer Random Forest Classifier

5.2 Tf-idf Vectorizer

5.2.1 Logistic Regression

Accuracy	0.66
Precision	0.67
Recall	0.66
F1 Score	0.65

Table 5.2.1.0: Result of Tf-idf Vectorizer Logistic Regression classifier

Confusion matrix

		Predicted Class	
		Hate	Not Hate
True Class	Hate	227	45
	Not Hate	123	105

Table 5.2.1.1: Confusion matrix of Tf-idf Vectorizer Logistic Regression classifier

Classification report

	precision	recall	f1-score	support
NO	065	0.83	0.73	272
YES	0.70	0.46	0.56	228

Table 5.2.1.2: Classification report of Tf-idf Vectorizer Logistic Regression classifier

Train Accuracy	0.879
Test Accuracy	0.664

Table 5.2.1.3: Train and Test Accuracy of Tf-idf Vectorizer Logistic Regression classifier

5.2.2 Multinomial Naive Bayes Classifier

Accuracy	0.71
Precision	0.71
Recall	0.71
F1 Score	0.70

Table 5.2.2.0: Result of Tf-idf Vectorizer Multinomial Naive Bayes Classifier

Confusion matrix

		Predicted Class	
		Hate	Not Hate
True Class	Hate	225	47
	Not Hate	100	128

Table 5.2.2.1: Confusion matrix of Tf-idf Vectorizer Multinomial Naive Bayes Classifier

Classification report

	precision	recall	f1-score	support
NO	0.69	0.83	0.75	272
YES	0.73	0.56	0.64	228

Table 5.2.2.2: Classification report of Tf-idf Vectorizer Multinomial Naive Bayes Classifier

Train Accuracy	0.895
Test Accuracy	0.706

Table 5.2.2.3: Train and Test Accuracy of Tf-idf Vectorizer Multinomial Naive Bayes Classifier

5.2.3 Linear SVM

Accuracy	0.68
Precision	0.70
Recall	0.68
F1 Score	0.66

Table 5.2.3.0: Result of Tf-idf Vectorizer Linear SVM Classifier

Confusion matrix

		Predicted Class	
		Hate	Not Hate
True Class	Hate	239	33
	Not Hate	129	99

Table 5.2.3.1 Confusion matrix of Tf-idf Vectorizer Linear SVM Classifier

Classification report

	precision	recall	f1-score	support
NO	0.65	0.88	0.75	272
YES	0.75	0.43	0.55	228

Table 5.2.3.2 Classification report of Tf-idf Vectorizer Linear SVM Classifier

Train Accuracy	0.857
Test Accuracy	0.676

Table 5.2.3.3 Train and Test of Tf-idf Vectorizer Linear SVM Classifier

5.2.4 Random Forest Classifier

Accuracy	0.68
Precision	0.69
Recall	0.68
F1 Score	0.67

Table 5.2.4.0: Result of Tf-idf Vectorizer Random Forest Classifier

Confusion matrix

	Predicted Class		
True Class		Hate	Not Hate
	Hate	227	45
	Not Hate	113	115

Table 5.2.4.1: Confusion matrix of Tf-idf Vectorizer Random Forest Classifier

Classification report

	precision	recall	f1-score	support
NO	0.67	0.83	0.74	272
YES	0.72	0.50	0.59	228

Table 5.2.4.2: Classification report of Tf-idf Vectorizer Random Forest Classifier

Train Accuracy	0.98
Test Accuracy	0.684

Table 5.2.4.3 Train and Test Accuracy of Tf-idf Vectorizer Random Forest Classifier

5.3 Evaluating Classifier methods with Difference N-gram values

5.3.1 countvectorizer

- **Unigram**

	Accuracy	precision	recall	f1-score
Logistic Regression	0.66	0.66	0.66	0.65
Multinomial Naive Bayes Classifier	**0.7**	**0.7**	**0.7**	**0.7**
Linear SVM	0.66	0.67	0.66	0.65
Random Forest Classifier	0.68	0.68	0.68	0.68

Table 5.3.1.13: Result of countvectorizer *unigram*

	Train Accuracy	Test Accuracy
Logistic Regression	0.9535	0.656
Multinomial Naive Bayes Classifier	**0.888**	**0.702**
Linear SVM	0.9535	0.662
Random Forest Classifier	0.98	0.68

Table 5.3.1.14: Result of countvectorizer *Train and Test Accuracy unigram*

- **Bigram**

	Accuracy	precision	recall	f1-score
Logistic Regression	0.66	0.66	0.66	0.66
Multinomial Naive Bayes Classifier	**0.71**	**0.71**	**0.71**	**0.71**
Linear SVM	0.66	0.66	0.66	0.65
Random Forest Classifier	0.67	0.68	0.67	0.67

Table 5.3.1.15:Result of countvectorizer *bigram*

	Train Accuracy	Test Accuracy
Logistic Regression	0.9695	0.656
Multinomial Naive Bayes Classifier	**0.897**	**0.712**
Linear SVM	0.9715	0.66
Random Forest Classifier	0.98	0.674

Table 5.3.1.16: Result of countvectorizer *Train and Test Accuracy bigram*

- **Trigram**

	Accuracy	precision	recall	f1-score
Logistic Regression	0.67	0.67	0.67	0.66
Multinomial Naive Bayes Classifier	**0.71**	**0.71**	**0.71**	**0.71**
Linear SVM	0.67	0.67	0.67	0.66
Random Forest Classifier	0.68	0.68	0.68	0.67

Table 5.3.1.17: Result of countvectorizer *trigram*

	Train Accuracy	Test Accuracy
Logistic Regression	0.97	0.666
Multinomial Naive Bayes Classifier	**0.8965**	**0.71**
Linear SVM	0.973	0.666
Random Forest Classifier	0.98	0.68

Table 5.3.1.18: Result of countvectorizer *Train and Test Accuracy trigram*

5.3.2 Tf-idf Vectorizer

- **Unigram**

	Accuracy	precision	recall	f1-score
Logistic Regression	0.66	0.67	0.66	0.64
Multinomial Naive Bayes Classifier	**0.70**	**0.71**	**0.70**	**0.70**
Linear SVM	0.68	0.69	0.68	0.67
Random Forest Classifier	0.68	0.69	0.68	0.67

Table 5.3.2.13: Result of Tf-idf Vectorizer unigram

	Train Accuracy	Test Accuracy
Logistic Regression	0.8895	0.66
Multinomial Naive Bayes Classifier	**0.9025**	**0.704**
Linear SVM	0.8875	0.682
Random Forest Classifier	0.98	0.680

Table 5.3.2.14: Summary Result of Train and Test Accuracy unigram

- **Bigram**

	Accuracy	precision	recall	f1-score
Logistic Regression	0.66	0.67	0.66	0.64
Multinomial Naive Bayes Classifier	**0.70**	**0.71**	**0.70**	**0.70**
Linear SVM	0.67	0.69	0.67	0.66
Random Forest Classifier	0.65	0.65	0.65	0.64

Table 5.3.2.15: Result of Tf-idf Vectorizer bigram

	Train Accuracy	Test Accuracy
Logistic Regression	0.89	0.66
Multinomial Naive Bayes Classifier	**0.9025**	**0.704**
Linear SVM	0.881	0.674
Random Forest Classifier	0.982	0.648

Table 5.3.2.16: Result of Tf-idf Vectorizer Train and Test Accuracy bigram

- **Trigram**

	Accuracy	precision	recall	f1-score
Logistic Regression	0.66	0.67	0.66	0.64
Multinomial Naive Bayes Classifier	**0.70**	**0.71**	**0.70**	**0.70**
Linear SVM	0.67	0.69	0.67	0.66
Random Forest Classifier	0.68	0.68	0.68	0.67

Table 5.3.2.17: Result of Tf-idf Vectorizer trigram

	Train Accuracy	Test Accuracy
Logistic Regression	0.89	0.66
Multinomial Naive Bayes Classifier	**0.9025**	**0.704**
Linear SVM	0.881	0.674
Random Forest Classifier	0.98	0.678

Table 5.3.2.18: Result of Tf-idf Vectorizer Train and Test Accuracy trigram

5.4 Summary of results

According to the above four classification method result extracted by using **countvectorizer**,

5.4.1 Summary Of countvectorizer

Logistic Regression

	Train Accuracy	Test Accuracy
unigram	0.972	0.658
Bigram	0.9695	0.656
Trigram	0.97	0.666

Table 5.4.1.3: Summary Result of countvectorizer Logistic Regression

Multinomial Naive Bayes Classifier

	Train Accuracy	Test Accuracy
unigram	**0.888**	**0.702**
Bigram	**0.897**	**0.712**
Trigram	**0.8965**	**0.71**

Table 5.4.1.6: Summary Result of countvectorizer Multinomial Naive Bayes Classifier

Linear SVM

	Train Accuracy	Test Accuracy
unigram	0.9535	0.662
Bigram	0.9715	0.66
Trigram	0.973	0.666

Table 5.4.1.9: Summary Result of countvectorizer Linear SVM Classifier

Random Forest Classifier

	Train Accuracy	Test Accuracy
unigram	0.98	0.68
Bigram	0.98	0.674
Trigram	0.98	0.68

Table 5.4.1.12: Summary Result of countvectorizer Random Forest Classifier

Here we examined all the best performing models from selected four models with different n-gram values and different min_DF values. So according to the Final Results **bigram and min _Df value is 3** is better classification values for each models.

	Train Accuracy	Test Accuracy

Logistic Regression	0.9695	0.656
Multinomial Naive Bayes Classifier	**0.897**	**0.712**
Linear SVM	0.9715	0.66
Random Forest Classifier	0.98	0.674

Table 5.4.1.13: Final Result of Countvactorizer

5.4.2 Summary Of Tf-idf Vectorizer

Logistic Regression

	Train Accuracy	Test Accuracy
unigram	0.8895	0.66
Bigram	0.8895	0.66
Trigram	0.89	0.66

Table 5.4.1.3: Summary Result of Tf-idf Vectorizer Logistic Regression

Multinomial Naive Bayes Classifier

	Train Accuracy	Test Accuracy
unigram	**0.9025**	**0.704**
Bigram	**0.9025**	**0.704**
Trigram	**0.9025**	**0.704**

Table 5.4.1.6: Summary Result of Tf-idf Vectorizer Multinomial Naive Bayes Classifier

Linear SVM

	Train Accuracy	Test Accuracy
unigram	0.8875	0.682
Bigram	0.881	0.674
Trigram	0.881	0.674

Table 5.4.1.9: Summary Result of Tf-idf Vectorizer Linear SVM Classifier

Random Forest Classifier

	Train Accuracy	Test Accuracy
unigram	0.98	0.680
Bigram	0.98	0.678
Trigram	0.98	0.678

Table 5.4.1.12: Summary Result of Tf-idf Vectorizer Random Forest Classifier

Here we examined all the best performing models from selected four models with different n-gram values and different min_DF values. So according to the Final Results **bigram and min _Df value is 3** is better classification values for each models.

	Train Accuracy	Test Accuracy
Logistic Regression	0.8895	0.66
Multinomial Naive Bayes Classifier	**0.9025**	**0.704**
Linear SVM	0.881	0.674
Random Forest Classifier	0.98	0.678

Table 5.4.1.13: Final Result of Tf-idf Vectorizer

5.4.3 Summary

In the random forest classifier method, when we evaluating those results we can see some overfitting the result on that classification methods. So I used the parameter tuning for the all classification algorithms especially for the random forest classifier I change the n_estimators value and random_state value then can see the some best results.

According to the above examined of Final Results of countvectorizer and *Tf-idf Vectorizer* feature extraction methods the Multinomial Naive Bayes Classifier model is better than other models with **bigram and min _Df value is 3**.

Multinomial Naive Bayes Classifier result with **bigram and min _Df value is 3**.

	Train Accuracy	Test Accuracy
countvectorizer	**0.897**	**0.712**
TFide	**0.9025**	**0.704**

Table 5.4.3: Final Result

Chapter 6: Conclusion and Future Works

6.1. Conclusion

In this project will look at difference ways to detect hate speech on social media and distinguish it from ordinary pornography. As the describe in the implementation chapter examine the social media comments by applying the difference feature extraction methods with difference key values to detect the hate speech.

6.2. Future work

As the initial step of the implementation, I tried to process the data using the soundex algorithm by pre-processed the data after generating the code for the words. I had modified the soundex algorithm phonetic for Romanized Sinhala but it is not a successful method. This is because the soundex algorithm for Romanized Sinhala has to be created. I will try to change the soundex algorithm for the Romanized Sinhala language as the future works.

In this study I only used countvectorizer and tfidf feature extraction methods and I will try to use cross validation for better results in future.

References

[1] "The Harm in Hate Speech — Jeremy Waldron." https://www.hup.harvard.edu/catalog.php?isbn=9780674416864 (accessed Jan. 24, 2020).

[2] "Liking violence: A study of hate speech on Facebook in Sri Lanka," *Centre for Policy Alternatives*, Sep. 24, 2014. https://www.cpalanka.org/liking-violence-a-study-of-hate-speech-on-facebook-in-sri-lanka/ (accessed Jan. 24, 2020).

[3] "Christchurch Shooting Live Updates: 49 Are Dead After 2 Mosques Are Hit - The New York Times." https://www.nytimes.com/2019/03/14/world/asia/new-zealand-shooting-updates-christchurch.html (accessed Jan. 24, 2020).

[4] "Hate Speech - ABA Legal Fact Check - American Bar Association." https://abalegalfactcheck.com/articles/hate-speech.html (accessed Jan. 24, 2020).

[5] "Policies - YouTube." https://www.youtube.com/about/policies/#community-guidelines (accessed Oct. 25, 2019).

[6] "Hateful conduct policy." https://help.twitter.com/en/rules-and-policies/hateful-conduct-policy (accessed Jan. 24, 2020).

[7] "Facebook." https://www.facebook.com/ (accessed Sep. 24, 2019).

[8] , Zhang Zuping1, N. D. G. *, Hanyurwimfura Damien2, and and Jun Long1, "A Lexicon-based Approach for Hate Speech Detection." .

[9] E. Riloff and J. Wiebe, "Learning extraction patterns for subjective expressions," in *Proceedings of the 2003 conference on Empirical methods in natural language processing* -, Not Known, 2003, vol. 10, pp. 105–112, doi: 10.3115/1119355.1119369.

[10] R. F. Martins, A. Pereira, and F. Benevenuto, "An Approach to Sentiment Analysis of Web Applications in Portuguese," in *Proceedings of the 21st Brazilian Symposium on Multimedia and the Web*, New York, NY, USA, 2015, pp. 105–112, doi: 10.1145/2820426.2820446.

[11] A. Joshi, "A Fall-back Strategy for Sentiment Analysis in Hindi: a Case Study," p. 6.

[12] Z. Waseem and D. Hovy, "Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter," in *SRW@HLT-NAACL*, 2016, doi: 10.18653/v1/n16-2013.

[13] "(PDF) Identifying Racist Social Media Comments in Sinhala Language Using Text Analytics Models with Machine Learning," *ResearchGate*. https://www.researchgate.net/publication/330469194_Identifying_Racist_Social_Media_Comments_in_Sinhala_Language_Using_Text_Analytics_Models_with_Machine_Learning (accessed Sep. 24, 2019).

[14] "Hatebase." https://hatebase.org/ (accessed Oct. 24, 2019).

[15] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, "Deep Learning for Hate Speech Detection in Tweets," *Proc. 26th Int. Conf. World Wide Web Companion - WWW 17 Companion*, pp. 759–760, 2017, doi: 10.1145/3041021.3054223.

[16] "Support-vector machine," *Wikipedia*. Dec. 01, 2019, Accessed: Jan. 24, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Support-vector_machine&oldid=928737848.

[17] "Logistic regression," *Wikipedia*. May 07, 2020, Accessed: May 15, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Logistic_regression&oldid=955290285.

[18] W. Koehrsen, “An Implementation and Explanation of the Random Forest in Python,” *Medium*, Aug. 31, 2018. https://towardsdatascience.com/an-implementation-and-explanation-of-the-random-forest-in-python-77bf308a9b76 (accessed May 15, 2020).
[19] “Applying Multinomial Naive Bayes to NLP Problems,” *GeeksforGeeks*, Jan. 11, 2019. https://www.geeksforgeeks.org/applying-multinomial-naive-bayes-to-nlp-problems/ (accessed Jan. 24, 2020).
[20] “6. Learning to Classify Text.html.” .
[21] “W17-1603.pdf.” .
[22] “Text processing and standardizing techniques used Tokenization In... | Download Scientific Diagram.” https://www.researchgate.net/figure/Text-processing-and-standardizing-techniques-used-Tokenization-In-Figure-2-the_fig2_323126520 (accessed Mar. 08, 2020).
[23] J. Brownlee, “A Gentle Introduction to the Bag-of-Words Model,” *Machine Learning Mastery*, Oct. 08, 2017. https://machinelearningmastery.com/gentle-introduction-bag-words-model/ (accessed Mar. 15, 2020).
[24] “Feature Extraction using TF-IDF algorithm - Hritik Attri - Medium.” https://medium.com/@hritikattri10/feature-extraction-using-tf-idf-algorithm-44eedb37305e (accessed Mar. 15, 2020).