CYBERBULLYING DETECTION BASED ON KAGGLE DATASET USING MACHINE LEARNING APPROACH

AFIEFAH HANNANI BINTI ABDUL HALIM

BACHELOR OF COMPUTER SCIENCE

(COMPUTER SYSTEM & NETWORKING) WITH HONORS

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name	:	AFIEFAH HANNANI BINTI ABDUL HALIM
Date of Birth	:	
Title	:	CYBERBULLYING DETECTION BASED ON KAGGLE DATASET USING MACHINE LEARNING APPROACH
Academic Session	:	SEMESTER I 2022/2023

I declare that this thesis is classified as:

□	CONFIDENTIAL	(Contains confidential information under the Official Secret Act 1997)*
□	RESTRICTED	(Contains restricted information as specified by the organization where research was done)*
☑	OPEN ACCESS	I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

(Student's Signature)	(Supervisor's Signature)
	MOHD FAIZAL BIN AB RAZAK
New IC/Passport Number	Name of Supervisor
Date: 08-02-2023	Date: 8/2/2023

NOTE: * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

THESIS DECLARATION LETTER (OPTIONAL)

Librarian,
Perpustakaan Universiti Malaysia Pahang,
Universiti Malaysia Pahang,
Lebuhraya Tun Razak,
26300, Gambang, Kuantan.

Dear Sir,

CLASSIFICATION OF THESIS AS RESTRICTED

Please be informed that the following thesis is classified as RESTRICTED for a period of three (3) years from the date of this letter. The reasons for this classification are as listed below.

Author's Name	AFIEFAH HANNANI BINTI ABDUL HALIM
Thesis Title	CYBERBULLYING DETECTION BASED ON KAGGLE DATASET USING MACHINE LEARNING APPROACH
Reasons	(i)
	(ii)
	(iii)

Thank you.

Yours faithfully,

DR. MOHD. FAIZAL BIN AB RAZAK
SENIOR LECTURER
FACULTY OF COMPUTER SYSTEMS & SOFTWARE ENGINEERING
UNIVERSITI MALAYSIA PAHANG
LEBUHRAYA TUN RAZAK, 26300 GAMBANG, KUANTAN
PAHANG DARUL MAKMUR
TEL : 09-549 2217 FAX : 09-549 2144

(Supervisor's Signature)

Date: 8/2/2023

Stamp:

Note: This letter should be written by the supervisor, addressed to the Librarian, *Perpustakaan Universiti Malaysia Pahang* with its copy attached to the thesis.

Universiti
Malaysia
PAHANG
Engineering • Technology • Creativity

SUPERVISOR’s DECLARATION

I hereby declare that I have checked this thesis and in my opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Bachelor of Computer Science in Computer System and Networking.

____

(Supervisor’s Signature)

Full Name : Mohd Faizal bin Ab Razak

Position : Senior Lecturer

Date : 8/2/2023

____

(Co-supervisor’s Signature)

Full Name :

Position :

Date :

UMP
Universiti
Malaysia
PAHANG
Engineering • Technology • Creativity

STUDENT’S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

____
(Student’s Signature)

Full Name : Afiefah Hannani binti Abdul Halim

ID Number : CA19084

Date : 08-02-2023

BEHAVIOUR ANALYSIS AMONG ADOLESCENTS AND CHILDREN FOR CYBERBULLYING BASED ON TWITTER AND KAGGLE DATASET

AFIEFAH HANNANI BINTI ABDUL HALIM

Thesis submitted in fulfillment of the requirements
for the award of the
Bachelor of Computer Science in Computer Systems and Networking

Faculty of Computer Systems and Software Engineering

UNIVERSITI MALAYSIA PAHANG

FEBRUARY 2023

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my project supervisor, Dr. Mohd Faizal bin Ab Razak, for their invaluable guidance, support, advices and encouragement in order to complete this thesis. Their expertise and knowledge in the field have been instrumental in shaping the direction and outcome of this research.

I am also grateful to the Universiti Malaysia Pahang (UMP), for providing access to the necessary resources and equipment during this thesis making. Without their support, this research would not have been possible.

Finally, I would like to thank my family and friends for their constant support and encouragement. Without them, this journey would not have been possible. A lot of thank you to all of you.

ABSTRAK

Pada masa kini, platform dalam talian digunakan terutamanya untuk berkomunikasi dan berhubung dengan orang ramai. Ia mengawal banyak kehidupan berdasarkan banyak aspek. Apabila dalam talian, tidak semua pengguna lebih suka apa yang mereka lihat. Mereka boleh bersuara tanpa memikirkan kesannya kepada seseorang. Daripada peristiwa ini, ia boleh membawa kepada aktiviti buli siber kerana semua pengguna bebas untuk melontarkan pendapat mereka di media sosial tanpa teragak-agak. Bukan itu sahaja, tetapi berdasarkan pertimbangan dalam talian, ia juga boleh menjejaskan kesihatan mental seseorang. Oleh itu, pengesanan buli siber menggunakan pendekatan Pembelajaran Mesin adalah dicadangkan. Kajian ini membentangkan perbandingan tiga algoritma Pembelajaran Mesin untuk pengesanan menggunakan aktiviti buli siber pada platform media sosial khususnya Twitter. Set data untuk melaksanakan algoritma akan diambil daripada tapak web sumber terbuka yang dipanggil Kaggle di mana ia akan digunakan untuk proses latihan dan ujian. Algoritma termasuk Mesin Vektor Sokongan (SVM), Naïve Bayes dan Decision Tree. Tujuan kajian ini adalah untuk melihat ketepatan algoritma dan membandingkannya. Algoritma tertinggi akan dipilih sebagai model dan algoritma terbaik yang boleh digunakan untuk mengesan teks tweet buli siber.

ABSTRACT

Nowadays, the online platform is primarily used to communicate and connect with people. It controlled many lives based on many aspects. When online, not all users prefer what they see. They can speak up without thinking about the effect on someone. From this event, it may lead to cyberbullying activity since all the users are free to throw their opinion on social media without hesitation. Not only that, but based on online judgment, it can also affect someone's mental health. Therefore, cyberbullying detection using a Machine Learning approach is suggested. This study presents the comparison of three Machine Learning algorithms for the detection using cyberbullying activity on social media platforms specifically Twitter. The dataset to perform the algorithm will be retrieved from an open-source website called Kaggle where it will be used for the training and testing process. The algorithms include Support Vector Machine (SVM), Naïve Bayes, and Decision Tree. The purpose of this study is to see the accuracy of the algorithms and compare it. The highest algorithm will be chosen as the best model and algorithm that can be used to detect cyberbullying tweet text.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

SVM	Support Vector Machine
DT	Decision Tree
NB	Naïve Bayes
KNN	K-nearest Neighbor
RBF	Radial Basis Function
TF-IDF	Term Frequency – Inverse Document Frequency

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Online communication is not only a common means to work anymore. It is also primarily used to communicate and connect with known and unknown individuals. These days the internet has controlled many lives based on many aspects. Many of them keep updating their daily life through social networking sites such as Instagram, Facebook, Twitter, TikTok, Snapchat, and many more as the way they communicate with others. Social media is now used in a variety of fields, including business, education, and good causes. Social networking is boosting the global economy by opening up a lot of new job prospects.

Social media offers many advantages, but it also has certain disadvantages. Through the usage of these platforms, malicious users engage in unethical and dishonest behaviour in an effort to offend others and harm their reputations. Cyberbullying has emerged as one of the most pressing social media problems recently. An electronic form of bullying or harassment is referred to as cyberbullying or cyber-harassment. Online bullying also refers to cyberbullying and cyber-harassment. Cyberbullying has increased in prevalence throughout time, especially among young people as the digital world and technology have expanded.

With the rise of social networking sites, there are also some bad side effects that the community faces. If 66% of children and teenagers had access to the internet at home in 2010, this figure is increasing year after year. According to national UK data, 21 million households which are 80% have internet connectivity in 2012, up from 19 million (77%) in 2011 [1]. Since users are free to throw their opinion on a certain post it could lead to makes someone become more insecure or not confident with themself. Not only that it also could affect one’s mental health. The negative aspect of young people's internet use is that they may bully or be bullied by others in cyberspace, also known as cyberbullying which can lead to short and long-term negative effects, including suicide or attempted suicide [1] .

Therefore, a method will be carried out as the prevention from the cyberbullying activities. The method that will be done is the cyberbullying text detection by using machine learning approach. As for this thesis, the text classification algorithm will be used to classify whether the text contains cyberbullying content or not. In this thesis, three classifiers will be used to detect the cyberbullying text which are Support Vector Machine (SVM), Naïve Bayes and Decision Tree. From these three techniques, we will see which method will produce the highest accuracy for detecting the cyberbullying text. To succeed in the thesis for this topic the tool that will be used to generate all the data is Python.

1.2 PROBLEM STATEMENT

Cyberbullying is a crime that normally happens on a digital device. Social media, online gaming platforms, or text messages are the common platforms for the bully to do this uneducated did. The way cyberbullying can happen is when a person is throwing a meaningful comment on someone’s post or insulting the victim’s look.

When online, not all users prefer what they see. They can speak up without thinking about the effect on someone. From this event, it may lead to cyberbullying activity since all the users are free to throw their opinion on social media without hesitation. Not only that, but based on online judgment, it can also affect someone’s mental health. Therefore, cyberbullying detection using a Machine Learning approach is suggested.

This study presents the comparison of three Machine Learning algorithms for the detection using cyberbullying activity on social media platforms specifically Twitter. The dataset to perform the algorithm will be retrieved from an open-source website called Kaggle where it will be used for the training and testing process. The algorithms include Support Vector Machine (SVM), Naïve Bayes, and Decision Tree. The purpose of this study is to see the accuracy of the algorithms and compare it. The highest algorithm will be chosen as the best model and algorithm that can be used to detect cyberbullying tweet text.

1.3 OBJECTIVE

The purpose of this research is to see how the accuracy for each algorithm. Here are the three objectives of this research.

i. To study the existing technique for cyberbullying detection.
ii. To evaluate cyberbully detection based on the accuracy from three classifiers, Support Vector Machine (SVM), Naïve Bayes and Decision Tree.
iii. To compare the classifier's accuracy output for detecting cyberbullying text.

1.4 SCOPE

Here is the scope of this research to get the wanted result:

Research scope:

i. This research will use the dataset from a media platform (Twitter).
ii. This research is also based on the dataset from Kaggle.
iii. This dataset will be divided into two parts for data train and data testing.
iv. The tool that will be used to compare the cyberbullying text is Python.

Methodology scope:

i. This research will use three classifiers for text classification and show the best accuracy among the three classifiers. (Naïve Bayes, Support Vector Machine (SVM) and Decision Tree)

1.5 REPORT ORGANIZATION

This thesis contains five chapters. Chapter one describes about the introduction of this research study which is Cyberbullying Detection Using Machine Learning Approach. This includes the introduction, problem statement, objectives and scope of the research.

Chapter two will discuss the literature review. In this chapter, we will see the comparison between 4 previous research papers. In this chapter, we will utilize many tools and techniques that were used in the previous research and the summarization from the findings.

As for chapter 3, it discusses on the methodology of the research. It covers the flow of the research, input, and output for the research, constrain and limitations during the research, and also the case study.

In Chapter 4, it discusses on the implementation based on the methodology that have been described in Chapter 3. Finally, Chapter 5 is about the conclusion which focus on research limitation, objective revisited, future work and the list of references for this research paper.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

This chapter will describe and compare three previous research papers about the detection of cyberbullying on Twitter.

2.2 PREVIOUS RESEARCH WORKS

This research paper is to propose a better way to detect cyberbullying activity from the Twitter platform by including an expanded keyword search from the other two references that the author used [2]. The study specifically attempts to identify tweets in which people share or expose their bullying experiences. In the data collection method, the authors gain data from Twitter's streaming API. From the data, the author will choose the English keyword that is related to cyberbullying which is called an enriched dataset. To trace the tweet that contains cyberbullying content, two models were used, human coded tweet and a machine learning approach using SVM and logistic regression. Results show that the machine learning approach is the best way to identify tweets that have cyberbullying elements.

The authors Nurrahmi & Nurjanah, 2018 treat the cyberbullying topic as the most important issue that must be known by Indonesians [3]. From this research paper, many methods are suggested by the other research that cannot be implemented for Indonesians. This is because of the language barrier and language used by the local communities. The study uses two platforms as the way to collect the data which is the data from 700 tweets under one account and a website named youswear.com. The first stage of the research is about the labelling system. In this stage, it talks about how the authors detect the inappropriate word which can lead to cyberbullying by using eight general rules. Since the data is raw and it is unlabelled, it will undergo a labelling process. Next is the detection of cyberbullying posts on Twitter. Three steps will be done in this stage. The first step is pre-processing. In this step, it will do the data cleaning, tokenization, and POS Tagging. After the pre-processing is done, the data will be divided into two parts which are 60% of data training and 40% of data testing. In this part, the SVM and KNN methods will be used. The result from the SVM and KNN methods will be compared to find the most accurate method for the research.

The research by Raihan is to analyze the different human behavior acts on social media [4]. This research is done by making a question paper with dome-related questions and distributing it to the universities and government offices around Bangladesh. After collecting all the data, the dataset will be processed. Then the dataset will apply the correlation Chi-Square test. Next, the open-source programming language “R” will be used to analyze the data. The expected result will be seen by looking at the p-value of the Chi-Square test.

This research paper is about the investigation of high school students' opinions and behavior about cyberbullying. The research is done by distributing 312 surveys to the random selection of classes. The method used by the researcher is by using the survey instrument by Willard (2004a) and her previous research paper [5]. This research gains a better output from the previous research and accurate data.

Table 2.2.1 Comparison and description of previous research papers

Elements	Previous Research 1	Previous Research 2	Previous Research 3	Previous Research 4
Research and author	Bullying discourse on Twitter: An examination of bully-related tweets using supervised machine learning [2].	Indonesian Twitter Cyberbullying Detection using Text Classification and User Credibility [3].	Human Behavior Analysis using Association Rule Mining Techniques [4].	Cyberbullying in High Schools: A Study of Students' Behaviors and Beliefes about This New Phenomenon [5].
Explanation	Investigates the sharing and disclosure of bullying events using Twitter data, including keywords that capture both face-to-face and cyberbullying incidents.	Discuss the way to detect cyberbullying activities based on text detection by using the text mining technique for text classification, the detection of cyberbullying actors by using the computation approach for user credibility management,	Analyze different human behavior acts on social media.	Investigates high school students' opinions and behavior about cyberbullying.

		and cyberbullying network visualization.		
Technique used	Human code tweets, Supervised Learning Machine (SVM and logistic regression).	Pre-process (Data cleaning, Tokenization, POS Tagging), K-nearest Neighbor (KNN), Support Vector Machine (linear and RBF kernel), Scikit-learn[2] (Python algorithm)	Open-source programming language “R”, Chi-Square test.	Survey instrument by Willard (2004a), own previous research paper.
Data	Twitter streaming API.	700 tweets from one account, youswear.com website.	Question paper.	312 survey was distributed to the random selection of classes.
Advantages	The accuracy of using machine learning in tracing cyberbullying on Twitter is higher than in human code tweets.	SVM can produce accurate data by using the RBF kernel.	The chi-square test is the effective method to gain the desired results.	Can gain more data from the survey and can improve previous research.
Disadvantages	Hard to classify the word related to cyberbullying.	SVM cannot perform well when using a	Need to use a large dataset to get a better	There will be students who do not return

	linear function.	result.	the survey form.

2.3 CONCLUSION

Based on these four previous research papers, the most suitable research paper to be as guidance is the research paper [3], Indonesian Twitter Cyberbullying Detection using Text Classification and User Credibility. Even though there are some different requirements for my research, the method and technique used can be upgraded by using a better coding algorithm.

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

The three classifiers will be applied to the research's goal. Each stage in a suggested study framework had to be carried out. The purpose of research methodology is to describe the direction of the investigation. Numerous research techniques have been employed often in earlier studies. The research framework and methodology that are appropriate for this study will be covered in this chapter. In this chapter, the techniques, method, and algorithm will all be covered in detail. The implementation of the framework suggested for this study will be covered in this chapter.

3.2 RESEARCH FRAMEWORK

The methodology of text classification for cyberbullying detection is shown as figure below:

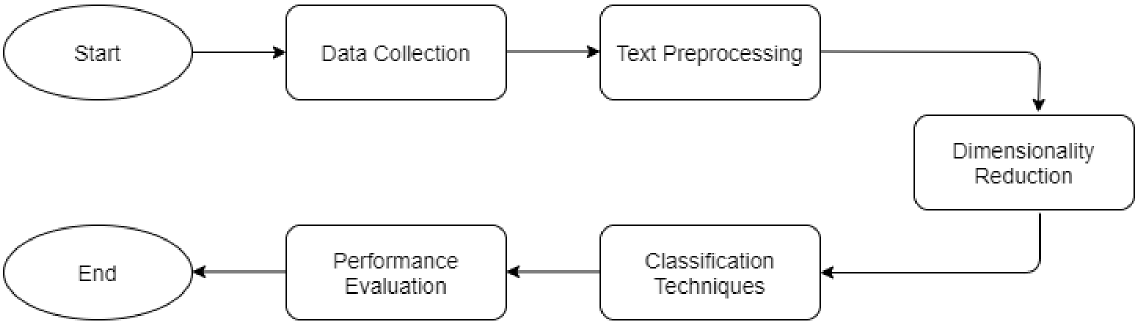


Figure 3.2.1.1Research Framework

3.2.1 Data collection

For the first phase of this methodology, we need to do the data collection. Since it is hard to pick the tweet text that have the cyberbullying element manually, the data will be collected from the open-source library which is Kaggle.

3.2.2 Data Pre-processing

There are many ways and techniques to do the data pre-processing. The techniques that will be used for the text classification are removing the unwanted and special character, remove the stop words, breaking the attach words, tokenization, lemmatizing and stemming. These steps are done because the raw text data may contain undesirable or unimportant text, which could make it difficult to interpret and analyse our results and prevent them from being accurate.

3.2.3 Dimensionality reduction

Techniques for reducing the number of input features in a dataset are referred to as dimensionality reduction. This phase of text mining is crucial. It condenses the set of features into a smaller form, which will increase the effectiveness of the learning process.

3.2.4 Classification techniques

In this part the dataset will be separated in to two parts for the data training and data testing. Here the classification process will be done using three classifiers which are Support Vector Machine (SVM), Naïve Bayes and Decision Tree classifier.

3.2.5 Performance evaluation

After it done with all the classification process, next it will do the evaluation from those three classifiers. In this part the accuracy from the classifiers will be compared and we will see the highest accuracy. The highest accuracy that obtained will be chosen as the best classifier that will be used to detect the cyberbullying text. Thus, we will evaluate the performance by comparing the accuracy for each classifier.

3.3 PROJECT REQUIREMENTS

3.3.1 Input

The data for this research is obtained by:

i. Twitter dataset
ii. Kaggle dataset

3.3.2 Output

The result of this thesis is that we will see the best accuracy among the three classifiers. (Naïve Bayes, Support Vector Machine (SVM) and Decision Tree)

3.3.3 Constraints and limitations

1. Data collected might need to undergo a long filtering process to get the desired data.
2. Need to analyze a big amount of data to get the best result.
3. The Support Vector Machine (SVM) classifier will take longer time to produce the accuracy and it will require the high amount of RAM.

3.3.4 Case Study

Online communication is no longer just a popular form of communication at work. It is also largely used to communicate and connect with known and unknown individuals. Nowadays, the internet has taken over many people's lives in a variety of ways.

Cyberbullying is a crime that occurs with digital technology. Cyberbullying commonly happens on social media, online gaming platforms, or SMS messaging. Cyberbullying can occur when someone makes a derogatory comment on another person's post or posts hatred content on someone.

When online, not all users prefer what they see. They can speak up without thinking about the effect on someone. From this event, it may lead to cyberbullying activity since all the users are free to throw their opinion on social media without hesitation. Not only that, but based on online judgment, it can also affect someone's mental health. Therefore, cyberbullying detection using a Machine Learning approach is suggested.

This study presents the comparison of three Machine Learning algorithms for the detection using cyberbullying activity on social media platforms specifically Twitter. The dataset to perform the algorithm will be retrieved from an open-source website called Kaggle where it will be used for the training and testing process. The algorithms include Support Vector Machine (SVM), Naïve Bayes, and Decision Tree. The purpose of this study is to see the accuracy of the algorithms and compare it. The highest algorithm will be chosen as the best model and algorithm that can be used to detect cyberbullying tweet text.

3.4 PROPOSED DESIGN

3.4.1 Flowchart

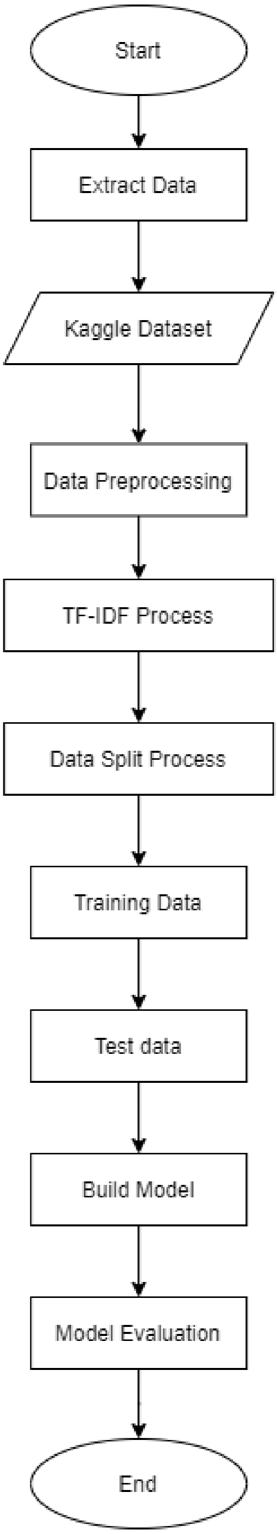


Figure 3.4.1.1 Research Framework

3.4.1.1 Data Extraction

The first stage of this research is extracting the data from the Kaggle dataset. Data extraction is the process of collecting or obtaining diverse forms of data from several sources, many of which may be poorly organised or unstructured. For this research, the data from the Kaggle dataset will be downloaded from the internet and then the data will be extracted to make the next process easier.

3.4.1.2 Data Pre-processing

This is the stage where the pre-processing of the text data is happening. The process called cleaning process. The pre-processing will start by removing the unwanted character such as punctuation, non-alphabets or any other character that is not included in language. After that, it will undergo the removing stop word process. Next, for the tokenization process. This process is done to split the sentence that we get from the data into words. The last step is lemmatizing and stemming process. The cleaning process is done to remove all the unwanted thing from the data.

3.4.1.3 TF-IDF process

TF-IDF is a short form of Term Frequency Inverse Document Frequency. NLP, or natural language processing, is the area of machine learning that deals with processing natural language data, such as text translation, sentiment analysis, user reviews, and user comments. The input data for all of the NLP-related issue statements is textual, which presents a big challenge. Before getting the accuracy for each algorithm all the dataset needs to be in numbering form. So, the use of TF-IDF in this part is to change all the words from the dataset to numbers. Every sentence had to be connected to a vector of numerical values, and this vector served as the model's input data. For this thesis, the method that is used from the TF-IDF process is the Tfidftransformer. This TF-IDF method will use the CountVectorizer to compute the word counts, then the Inverse Document Frequency value and TF-IDF score

3.4.1.4 Data Split, Data Testing and Data Train

For the last process of this research, the data will be split into two, training data and test data. The whole dataset from Kaggle will be split into two parts where 80% of the data is for data training and 20% of the data is for data testing. From the data train and data testing process, it will continue to the build model for Naïve Bayes algorithm, Support Vector Machine algorithm and Decision Tree algorithm and the evaluation process. Here, the accuracy from each classifier will be obtain and compared.

3.4.2 Text Classification Algorithm

3.4.2.1 Support Vector Machine (SVM)

Based on (Li, 2018), said that SVM is one of the best classifiers that we can use for the classification algorithm. SVM works by mapping data to a high-dimensional feature space in order to categorise data points that are otherwise not linearly separable. A separator between the categories is discovered, and the data are processed so that the separator may be drawn as a hyperplane. Below are the steps on how SVM is happening.

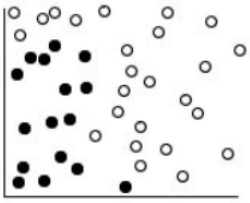


Figure 3.4.2.1 Original dataset for SVM

Based in figure above we can see that it is the original dataset. The dataset is from two different categories.

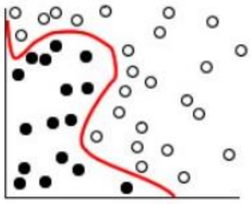


Figure 3.4.2.2 Curve separator

Next when the data is already collected, then it will be separated with a curve line since the two data mixed.

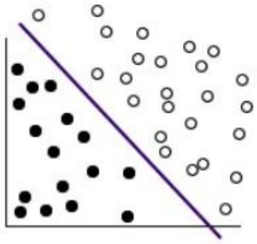


Figure 3.4.2.3 Hyperplane representation

After the data had been separated with a curve separator, it will transform and produce a hyperplane by using kernel function.

3.4.2.2 Naïve Bayes

Naïve Bayes is a simple algorithm that classifies text based on the likelihood of events occurring. This procedure is based on the Bayes theorem, which aids in determining the conditional probability of events that occurred based on the odds of each event occurring. So, for a given text, we calculate the probability of each tag and output the tag with the highest probability. To understand more about this algorithm, several steps need to be done.

The first step that we need to do is we need to confirm they type of feature engineering that we wanted to use. For an example a word frequencies feature will be used. This feature engineering is used when the data only have text. The way how this feature will work is it will disregard the word order and the sentence structure, treating each data as a collection of the words it contains. Next, we must convert the probability we wish to compute into something that can be computed using word frequencies. We will use some basic probability principles as well as Bayes' Theorem to do this.

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

Figure 3.6 Bayes’ Theorem

Now comes the naive part where we presume that each word in a phrase is independent of the others. This indicates that we are now examining individual words rather than complete phrases. Below is the example on how this part is done.

$$P(a\,very\,close\,game) = P(a) \times P(very) \times P(close) \times P(game)$$

Figure 3.7 Implementation of naïve process

The last step for Naïve Bayes algorithm is it will calculate the probability for each word and see the highest probability among the words and the highest probability will be choose as the result.

3.4.2.2 Decision tree

A decision tree is a supervised learning technique that is ideal for classification issues since it can organize classes precisely. It works like a flow chart, splitting data points into two related categories at a time, starting with the "tree trunk" and progressing to "branches" and "leaves," where the categories become more finitely similar. This results in the formation of categories inside categories, allowing for organic classification with minimal human supervision. Below is the example on how this algorithm is implemented.

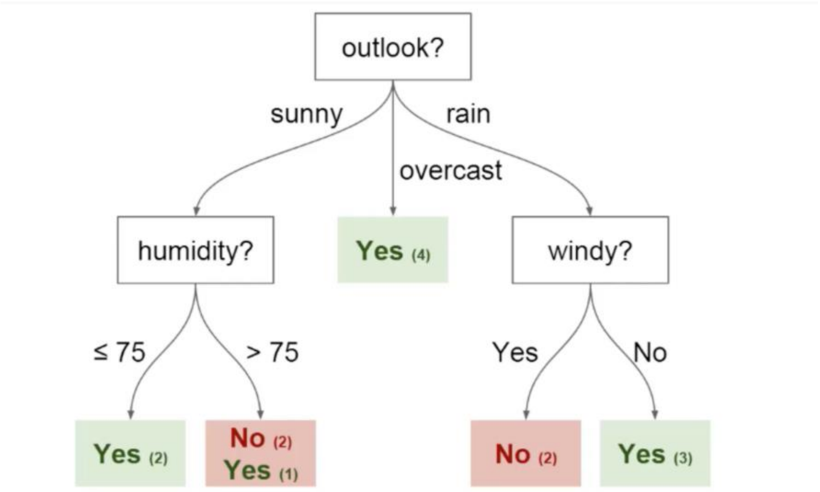
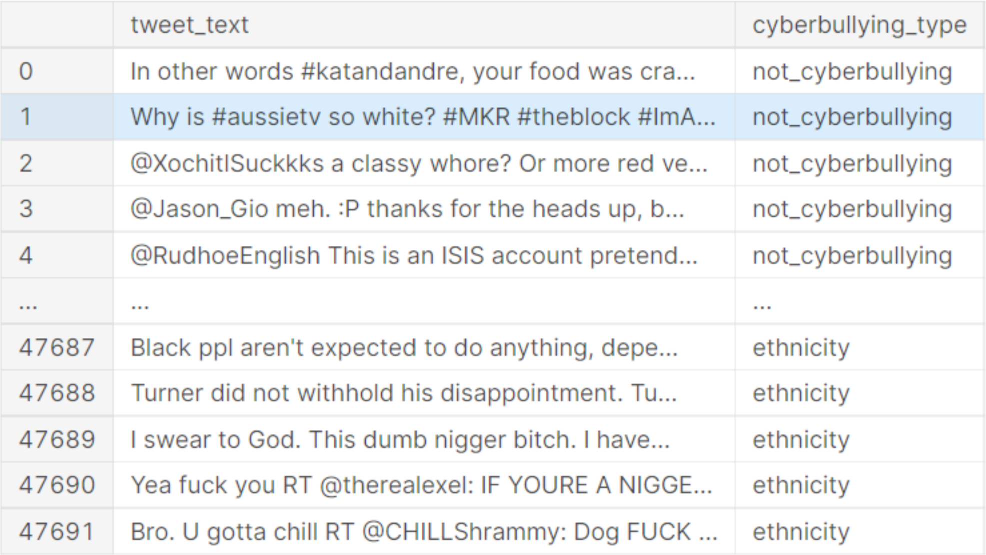


Figure 3.4.2.4 Decision tree implementation [6]

3.5 DATA DESIGN

The data that will be used for this thesis is a dataset from Kaggle titled Twitter Cyberbullying Text Classification from Feri Haldi Tanjung. In that dataset, there are more than 47000 tweets from Twitter users that possibly contain the cyberbullying issue. The dataset is categorized into several categories such as age, ethnicity, gender, religion, other types of cyberbullying, and not cyberbullying. Each category of the dataset contains 8000 tweets. Below is the figure on what this dataset looks like.



	tweet_text	cyberbullying_type
0	In other words #katandandre, your food was cra...	not_cyberbullying
1	Why is #aussietv so white? #MKR #theblock #ImA...	not_cyberbullying
2	@XochitlSuckkks a classy whore? Or more red ve...	not_cyberbullying
3	@Jason_Gio meh. :P thanks for the heads up, b...	not_cyberbullying
4	@RudhoeEnglish This is an ISIS account pretend...	not_cyberbullying
...	...	...
47687	Black ppl aren't expected to do anything, depe...	ethnicity
47688	Turner did not withhold his disappointment. Tu...	ethnicity
47689	I swear to God. This dumb nigger bitch. I have...	ethnicity
47690	Yea fuck you RT @therealexel: IF YOURE A NIGGE...	ethnicity
47691	Bro. U gotta chill RT @CHILLShrammy: Dog FUCK ...	ethnicity

Figure 3.4.2.1 Presentation of the dataset [7]

After the data has been collected, then all the tweets will be categorized and presented using the pie chart. The pie chart shows that, the percentage

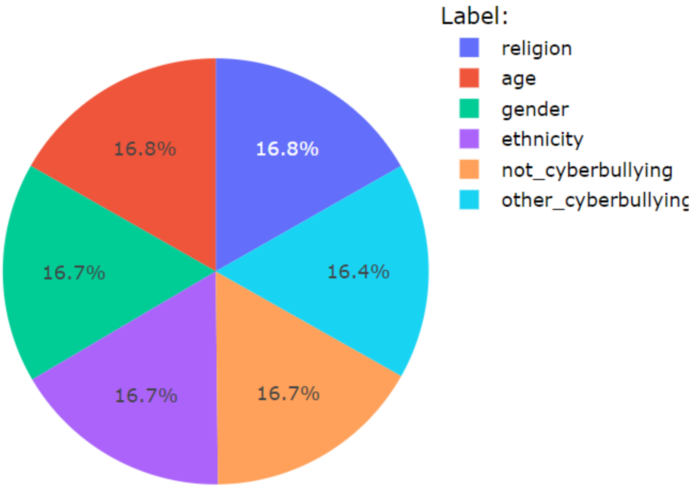


Figure 3.4.2.2 Percentage of each category of cyberbullying type [7]

After categorizing the tweets, then it will use Python code to determine the most frequent username that appears, most frequent hashtags that appear, emojis, character length, toxic level in each tweet, most toxic tweet for each category, and most frequent word that appears.

3.6 PROOF OF INITIAL CONCEPT

3.6.1 Evidence of early work and comparative analysis

Based on the previous part, this research stated that the algorithm that will be used is the text classification algorithm with three classifiers which are the Support Vector Machine (SVM) classifier, decision tree classifier, and Naïve Bayes classifier. So, for this part, I will show the previous study that use the text classification algorithm with the three classifiers.

The research paper that will be used in this part is from Talpur, B. A., & O'Sullivan, D. [8] with title Multi-class imbalance in text classification: A feature engineering approach to detect cyberbullying in Twitter. One of the objectives from this research paper is to develop a machine learning classifier to classify the tweets as non-cyberbullied, low, medium, or high level of cyberbullied. Since the method used in this research paper is much more advanced than supervised machine learning, the writer concludes there is more method that we can use to get the er results for classifying text.

The second research paper is from Noviantho, S. I., & Ashianti, L. [9] with title Cyberbullying Classification using Text Mining. This research uses two methods to detect the cyberbullying message which are Support Vector Machine (SVM) and Naïve Bayes method. Based on the conclusion that I get from this research is that the researcher stated that SVM kernel produce the most accurate result for detecting the cyberbullying text with 99.04% accuracy while Naïve Bayes is 96.98% accuracy.

3.7 PROOF OF INITIAL CONCEPT

3.7.1 Design

This part is based on the research paper from [10] with title Comparison of Naïve Bayes, Support Vector Machine, Decision Trees and Random Forest on Sentiment Analysis.

To gain the best result, this research paper measures each classifier based on four elements which are accuracy, precision, recall and F1 score. Based on the activity that had been done, the writer concluded that Support Vector Machine (SVM) is the best classifier. This is because the classifier is the most accurate classifier and have the highest value for each of the elements that they measure.

As conclusion we can know that the Support Vector Machine (SVM) is the best classifier that we can use.

3.7.2 Usability

In this research the method that being used is the text classification algorithm. Here are the reasons why text classification algorithm is the best method. The first reason why is text classification algorithm is the best method to classify things is because of the scalability. Manually analysing and arranging is time-consuming and inaccurate. Machine learning can evaluate millions of surveys, comments, emails, and other data sets automatically for a fraction of the cost, typically in only a few minutes. Text classification technologies can be scaled to meet the needs of any business, large or small.

Next, is about the real-time analysis. There are important situations that businesses must detect as soon as possible. Machine learning text classification can track your brand mentions in real-time, allowing you to detect vital information and act quickly.

Lastly is about the consistency of the criteria. Because of distractions, exhaustion, and boredom, human annotators make mistakes while classifying text data, and human subjectivity provides inconsistent standards. Machine learning, on the other hand, looks at all data and results through the same lens and criterion. Once correctly trained, a text classification model performs with unrivalled precision. Those are the reasons why text classifications are the best method for classifying something.

CHAPTER 4

IMPLEMENTATION, RESULTS AND DISCUSSION

4.1 INTRODUCTION

As mentioned before, three algorithms will be used to check the accuracy for detecting cyberbullying text from the cyberbullying dataset. To detect the cyberbullying words from the tweet text, Python programming language will be used to find the accuracy among those three algorithms which are Support Vector Machine (SVM), Naïve Bayes, and Decision Tree algorithm. From these three algorithms, we will compare the accuracy and the highest accuracy will be chosen as the best algorithm that can be used as the cyberbullying predictor.

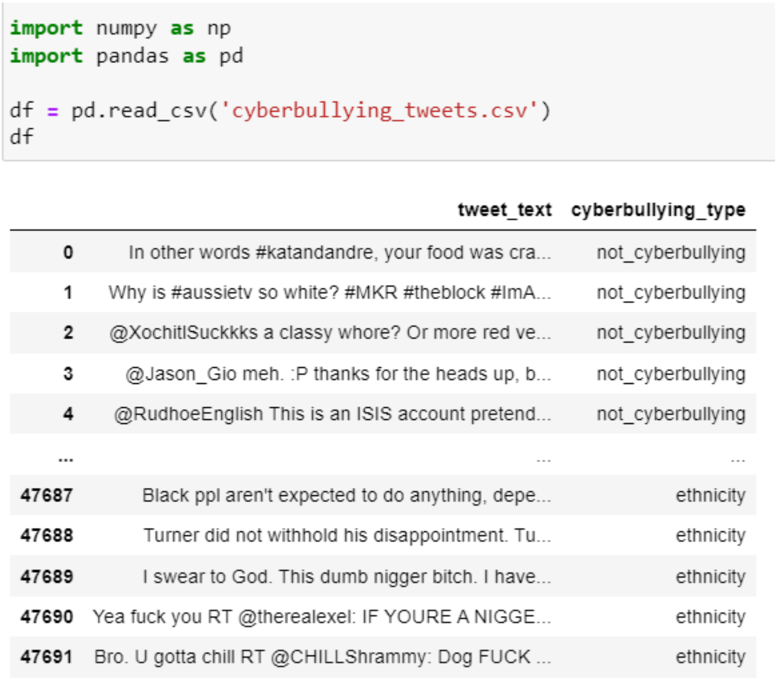
4.2 IMPLEMENTATION

Here is the methodology of the implementation:

i. Analysing data.
ii. Data Pre-processing.
iii. TF-IDF process.
iv. Data Training and Data Testing.

4.2.1 Analysing data

The first step before the accuracy for Support Vector Machine (SVM), Naïve Bayes, Decision Tree produced is the dataset that collected need to be analysed. This is done to see the information or details about the dataset. To do that, the data need to be upload first. Figure 4.1 shows the process of loading the data and the description that the dataset has. The figure shows the tweet text that is collected from Twitter and the type of cyberbullying. To present the details about the dataset, the df function is used.

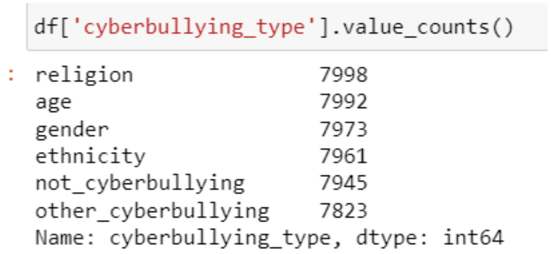


```
import numpy as np
import pandas as pd

df = pd.read_csv('cyberbullying_tweets.csv')
df
```

	tweet_text	cyberbullying_type
0	In other words #katandandre, your food was cra...	not_cyberbullying
1	Why is #aussietv so white? #MKR #theblock #ImA...	not_cyberbullying
2	@XochitlSuckkks a classy whore? Or more red ve...	not_cyberbullying
3	@Jason_Gio meh. :P thanks for the heads up, b...	not_cyberbullying
4	@RudhoeEnglish This is an ISIS account pretend...	not_cyberbullying
...	...	...
47687	Black ppl aren't expected to do anything, depe...	ethnicity
47688	Turner did not withhold his disappointment. Tu...	ethnicity
47689	I swear to God. This dumb nigger bitch. I have...	ethnicity
47690	Yea fuck you RT @therealexel: IF YOURE A NIGGE...	ethnicity
47691	Bro. U gotta chill RT @CHILLShrammy: Dog FUCK ...	ethnicity

Figure 4.2.1.1 Dataset details

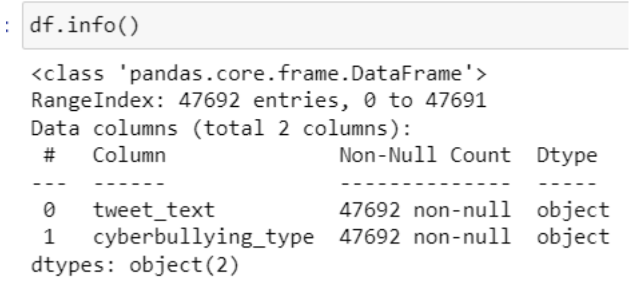
To get the accurate results for each algorithm, all the details about the dataset need to be analysed. As shown below, Figure 4.2 shows all the cyberbullying types with the total of tweet text for each cyberbullying types while Figure 4.3 shows all the information about the dataset.



```
df['cyberbullying_type'].value_counts()
```

```
religion               7998
age                    7992
gender                 7973
ethnicity              7961
not_cyberbullying      7945
other_cyberbullying    7823
Name: cyberbullying_type, dtype: int64
```

Figure 4.2.1.2 Total counts of each cyberbullying types.



```
df.info()
```

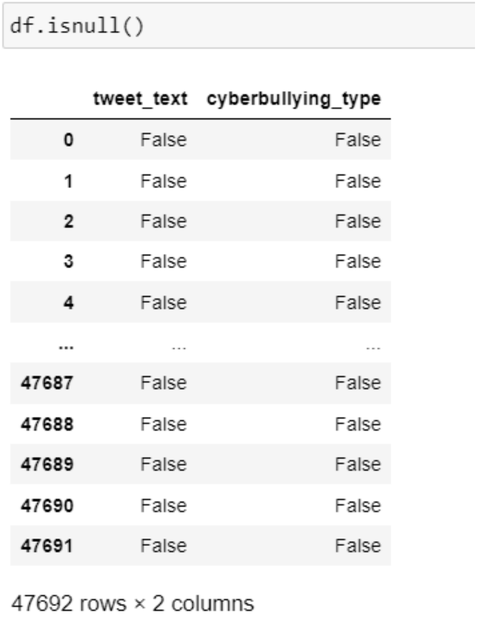
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 47692 entries, 0 to 47691
Data columns (total 2 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   tweet_text          47692 non-null  object
 1   cyberbullying_type  47692 non-null  object
dtypes: object(2)
```

Figure 4.2.1.3 The information of dataset.

4.2.2 Data pre-processing

i. Checking null data from the dataset.

When there is an empty space in the dataset, it is referred to as null and denotes that the field's value is unknown. As for the dataset that we obtain from the public library (Kaggle), it might have some defect since the data is reused by many people. To prevent any missing data from the dataset, checking the null is a must. The result from checking the null data is shown as below.



```
df.isnull()
```

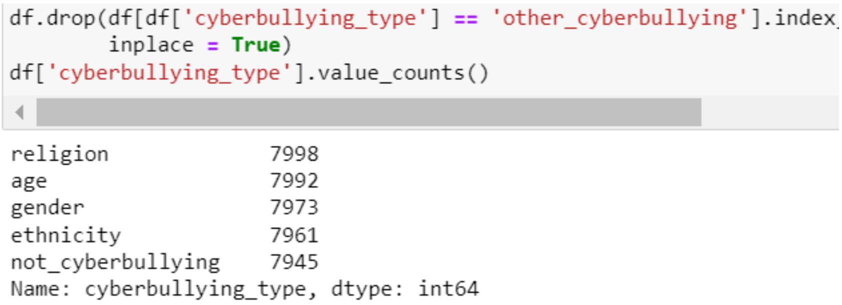
	tweet_text	cyberbullying_type
0	False	False
1	False	False
2	False	False
3	False	False
4	False	False
...	...	...
47687	False	False
47688	False	False
47689	False	False
47690	False	False
47691	False	False

47692 rows × 2 columns

Figure 4.2.2.1 Checking null data from the dataset

ii. Remove one of cyberbullying types.

As shown in Figure 4.2, there is one of the cyberbullying types that may cause confusion when producing the accuracy for the algorithms which is other_cyberbullying type. To prevent confusion from happening, the data will be removed so that the algorithms will produce the best accuracy.



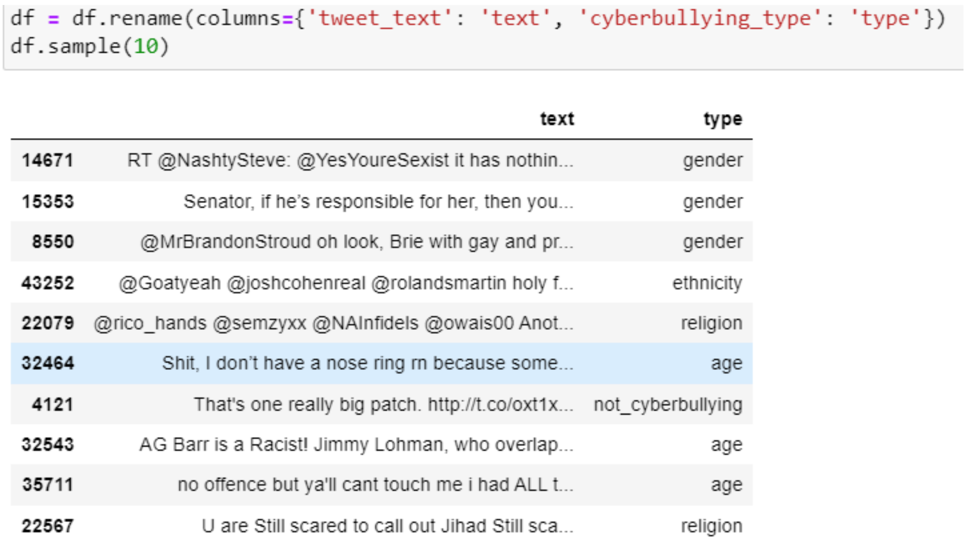
```
df.drop(df[df['cyberbullying_type'] == 'other_cyberbullying'].index,
        inplace = True)
df['cyberbullying_type'].value_counts()
```

```
religion               7998
age                    7992
gender                 7973
ethnicity              7961
not_cyberbullying      7945
Name: cyberbullying_type, dtype: int64
```

Figure 4.2.2.2 Remove one of cyberbullying types

iii. Renaming attributes.

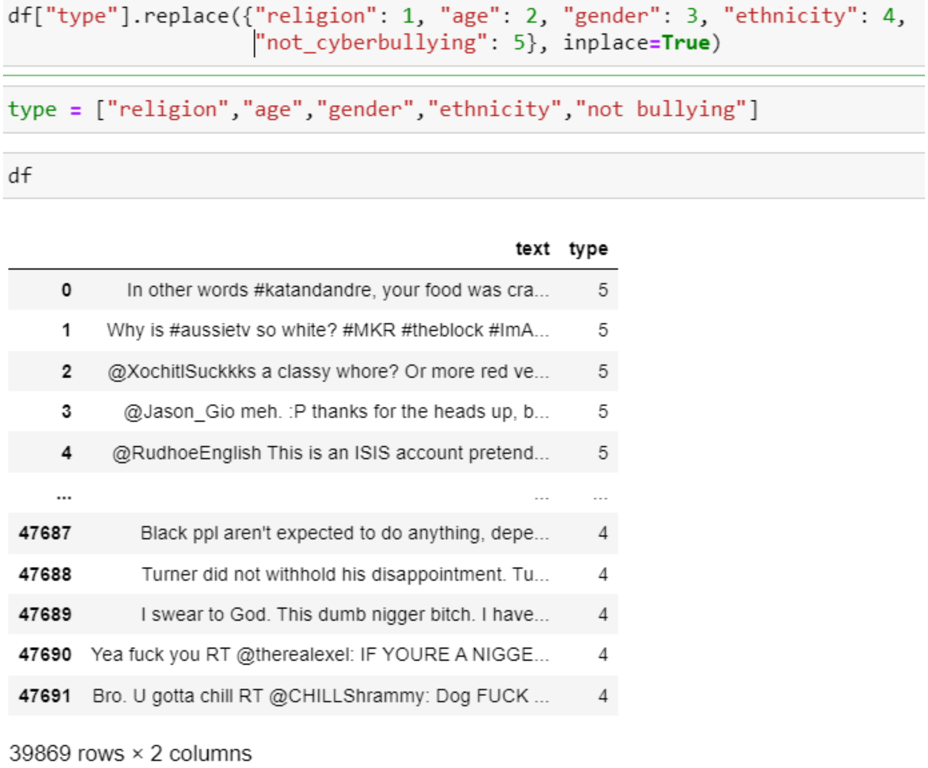
This process is done to simplify the existing attributes. From this step, it will make the attribute can easily being trace if there is error later. Since all the data from the dataset is in word form, it is impossible to generate the accuracy using the words without changing it to numbers. So, the step in Figure 4.6 is done to replace the cyberbullying types in word form to numbering form. Below are the steps for renaming the attributes.



```
df = df.rename(columns={'tweet_text': 'text', 'cyberbullying_type': 'type'})
df.sample(10)
```

	text	type
14671	RT @NashtySteve: @YesYoureSexist it has nothin...	gender
15353	Senator, if he's responsible for her, then you...	gender
8550	@MrBrandonStroud oh look, Brie with gay and pr...	gender
43252	@Goatyeah @joshcohenreal @rolandsmartin holy f...	ethnicity
22079	@rico_hands @semzyxx @NAInfidels @owais00 Anot...	religion
32464	Shit, I don't have a nose ring rn because some...	age
4121	That's one really big patch. http://t.co/oxt1x...	not_cyberbullying
32543	AG Barr is a Racist! Jimmy Lohman, who overlap...	age
35711	no offence but ya'll cant touch me i had ALL t...	age
22567	U are Still scared to call out Jihad Still sca...	religion

Figure 4.2.2.3 Renaming attributes



```
df["type"].replace({"religion": 1, "age": 2, "gender": 3, "ethnicity": 4,
                    "not_cyberbullying": 5}, inplace=True)
```

```
type = ["religion","age","gender","ethnicity","not bullying"]
```

```
df
```

	text	type
0	In other words #katandandre, your food was cra...	5
1	Why is #aussietv so white? #MKR #theblock #ImA...	5
2	@XochitlSuckkks a classy whore? Or more red ve...	5
3	@Jason_Gio meh. :P thanks for the heads up, b...	5
4	@RudhoeEnglish This is an ISIS account pretend...	5
...	...	...
47687	Black ppl aren't expected to do anything, depe...	4
47688	Turner did not withhold his disappointment. Tu...	4
47689	I swear to God. This dumb nigger bitch. I have...	4
47690	Yea fuck you RT @therealexel: IF YOURE A NIGGE...	4
47691	Bro. U gotta chill RT @CHILLShrammy: Dog FUCK ...	4

39869 rows × 2 columns

Figure 4.2.2.4 Replacing text form cyberbullying type to numbering form

iv. Data cleaning process.

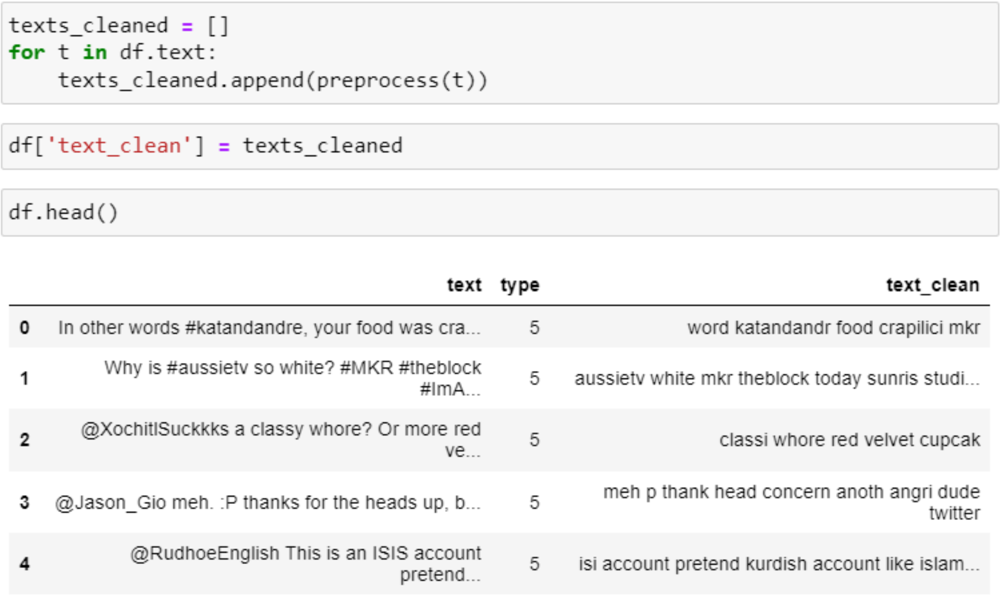
Table 4.2.1 Details of data cleaning process

Types of text cleaning	Description
• Change all text to lower case	• In computer, the lower case and upper-case sentence are handled differently. The reason of changing all the words to lower case is to make the machine easy to read the words. For example, the word “Lion” and “lion” for instance, receive distinct treatment from the computer. In order to avoid these issues, the text must be written in the same case, with lower case being the most desired.
• Removing punctuation and special character.	• For this process all the 32 punctuations such as “!#$_@” [1] will be removed and it will replace by the empty string. Same goes with the special character.
• Remove stop words.	• The words like “the, they, where, the, a and an” are examples of stop words. The main reason of removing these kinds of words is because it has no valuable

	information in a text.
• Remove ascii characters	• Ascii characters is a combination of words and numbers. We can express the ascii character as gta78 for example. By having this kind of character in the text, it will make the computer hard to understand the words and it is difficult for the computer to process. To make the machine work become smooth, the ascii character will be remove from the text and it will be replaced by the empty string.
• Remove contractions	• Words like can’t, don’t and I’ll are some of the examples of contractions. It is actually a word groups that have had letters removed and been replaced with an apostrophe.
• Tokenization	• Tokenization is a pre-processing the will split a sentence into part by part or also known as token. To make it more clear here is the example . • Full sentence: ‘My name is Afiefah’ • Tokenize sentence: [‘My’, ‘name’, ‘is’, ‘Afiefah’]

• Lemmatization & Stemming	• Lemmatization is a process where it will convert a word into a root form while for stemming process it will remove the last few characters from the word. • The difference between these two processes is that, lemmatization process it will look at the whole context of the word before convert it to root word while for stemming process it just cut off the last few words such as ‘ing’ to make it become a root form. Below is the example for these two processes. • Word: Caring Lemmatization: Care Stemming: Car

Figure 4. are the results obtained from the data cleaning process.



```
texts_cleaned = []
for t in df.text:
    texts_cleaned.append(preprocess(t))
```

```
df['text_clean'] = texts_cleaned
```

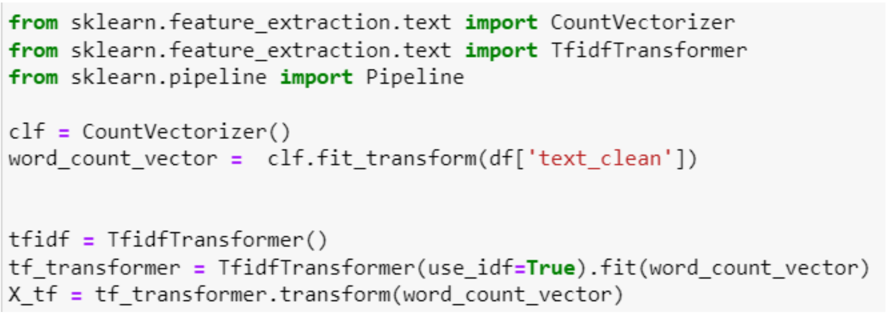
```
df.head()
```

	text	type	text_clean
0	In other words #katandandre, your food was cra...	5	word katandandr food crapilici mkr
1	Why is #aussietv so white? #MKR #theblock #ImA...	5	aussietv white mkr theblock today sunris studi...
2	@XochitlSuckkks a classy whore? Or more red ve...	5	classi whore red velvet cupcak
3	@Jason_Gio meh. :P thanks for the heads up, b...	5	meh p thank head concern anoth angri dude twitter
4	@RudhoeEnglish This is an ISIS account pretend...	5	isi account pretend kurdish account like islam...

Figure 4.2.2.5 Data cleaning result

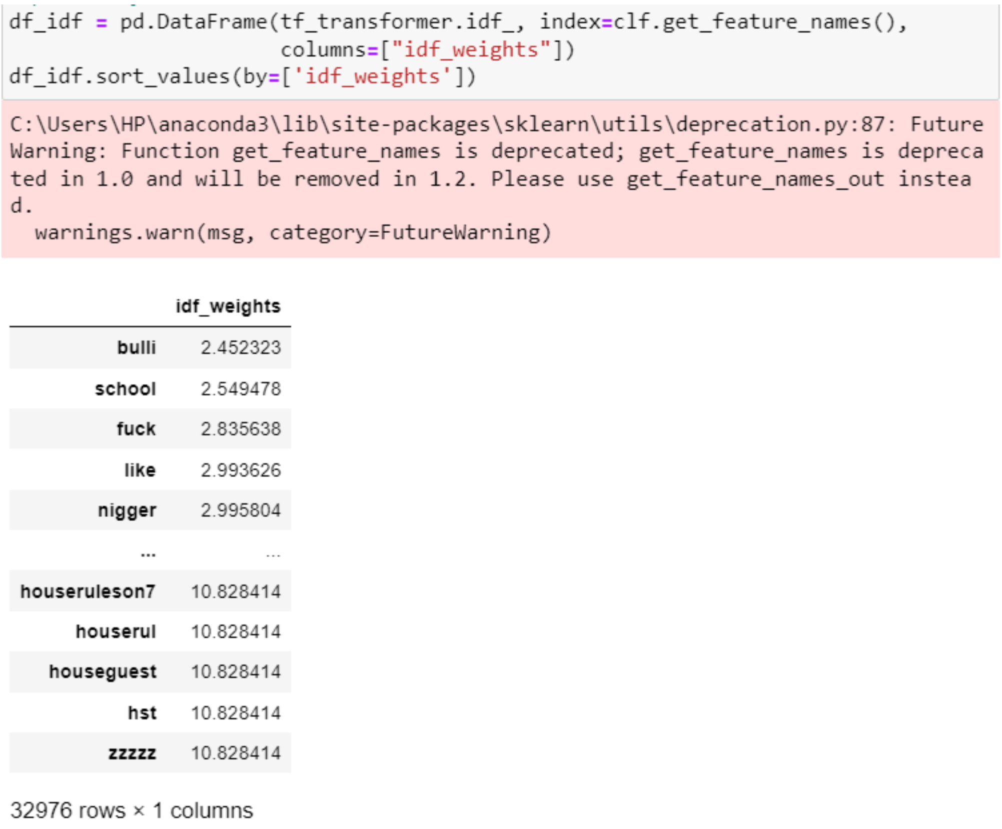
4.2.3 TF-IDF process

TF-IDF is a short form of Term Frequency Inverse Document Frequency. NLP, or natural language processing, is the area of machine learning that deals with processing natural language data, such as text translation, sentiment analysis, user reviews, and user comments. The input data for all of the NLP-related issue statements is textual, which presents a big challenge. Before getting the accuracy for each algorithm all the dataset needs to be in numbering form. So, the use of TF-IDF in this part is to change all the words from the dataset to numbers. Every sentence had to be connected to a vector of numerical values, and this vector served as the model's input data. For this thesis, the method that is used from the TF-IDF process is the Tfidftransformer. This TF-IDF method will use the CountVectorizer to compute the word counts, then the Inverse Document Frequency value and TF-IDF score. Below is the way of using Tfidftransformer for the cyberbullying dataset.



```python
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.pipeline import Pipeline

clf = CountVectorizer()
word_count_vector =  clf.fit_transform(df['text_clean'])


tfidf = TfidfTransformer()
tf_transformer = TfidfTransformer(use_idf=True).fit(word_count_vector)
X_tf = tf_transformer.transform(word_count_vector)
```

Figure 4.2.3.1 Tfidftransformer method application



```python
df_idf = pd.DataFrame(tf_transformer.idf_, index=clf.get_feature_names(),
                      columns=["idf_weights"])
df_idf.sort_values(by=['idf_weights'])
```

```
C:\Users\HP\anaconda3\lib\site-packages\sklearn\utils\deprecation.py:87: Future
Warning: Function get_feature_names is deprecated; get_feature_names is depreca
ted in 1.0 and will be removed in 1.2. Please use get_feature_names_out instea
d.
  warnings.warn(msg, category=FutureWarning)
```

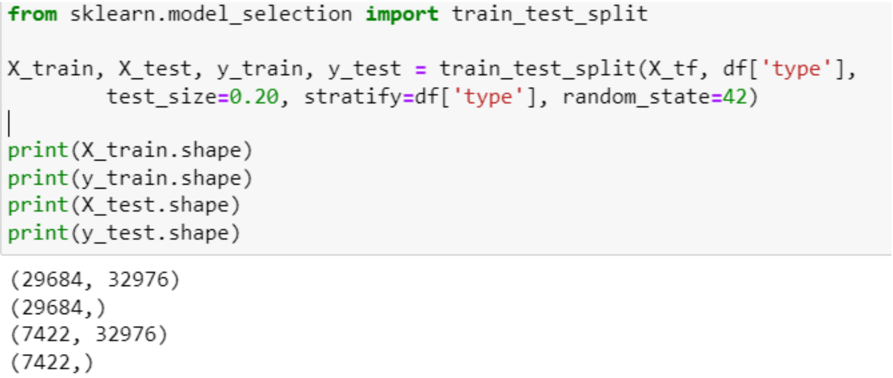
	idf_weights
bulli	2.452323
school	2.549478
fuck	2.835638
like	2.993626
nigger	2.995804
...	...
houseruleson7	10.828414
houserul	10.828414
houseguest	10.828414
hst	10.828414
zzzzz	10.828414

32976 rows × 1 columns

Figure 4.2.3.2 Result of TF-IDF process

4.2.4 Splitting data for data training and data testing process

The dataset that had undergo all the pre-processing is now ready to do the data training and data testing process. The dataset will be divided into two parts where 80% of the data from the whole dataset will do the data training while the other 20% of the dataset will undergo the testing process. The expected result, which includes both input and expected output, is included in the training data while the testing data is used to determine whether the trained model works well on unobserved data. Once it has received thorough training, the data is used to predict the outcome.



```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X_tf, df['type'],
        test_size=0.20, stratify=df['type'], random_state=42)

print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)
```

```
(29684, 32976)
(29684,)
(7422, 32976)
(7422,)
```

Figure 4.2.4.1 Data split, train and test

4.3 RESULTS AND DISCUSSION

In this part all the results obtain from each classifier will be he shown here. To get the best accuracy for each algorithm, the correct input is needed in order to get the desired output. These algorithms will produce the accuracy from each cyberbullying category which are not_cyberbullying, gender, age, religion and ethnicity. All the results obtain from the Naïve Bayes algorithm, Support Vector Machine (SVM) algorithm and Decision Tree algorithm will be discussed.

4.3.1. Most frequent words in each cyberbullying type

Here are the results that obtain from the pre-processing process. All the clean data that has undergo the pre-processing process will be listed by types of cyberbullying first. Then, from the list, the word cloud will be generated to show the most frequents words that have in that particular cyberbullying type. The figures below show the word cloud that had been generated.

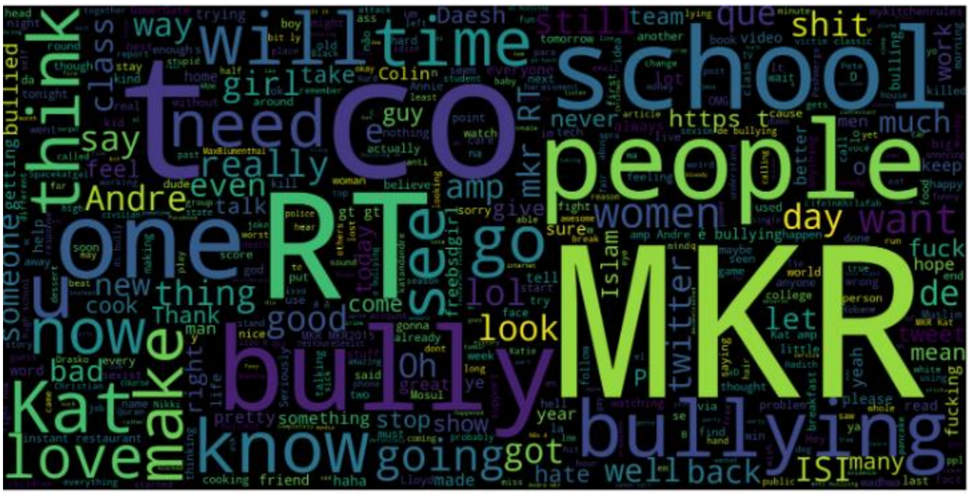


Figure 4.3.1.1 Frequent words of not_cyberbullying cyberbullying type

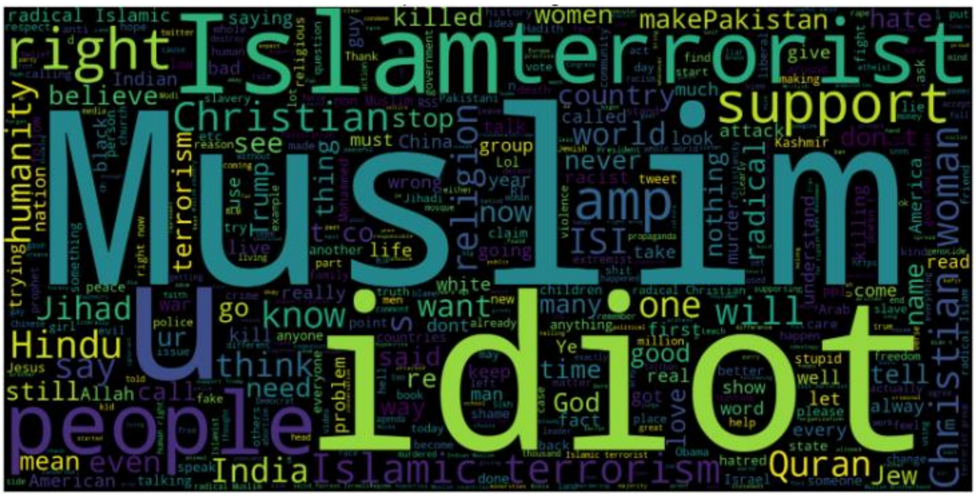


Figure 4.3.1.2 Frequent words in religion cyberbullying type

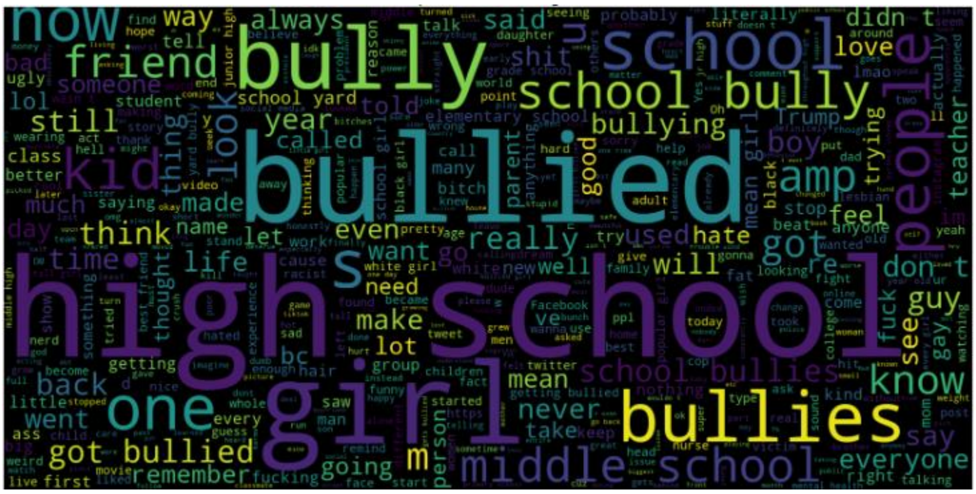


Figure 4.3.1.3 Frequent words in age cyberbullying type

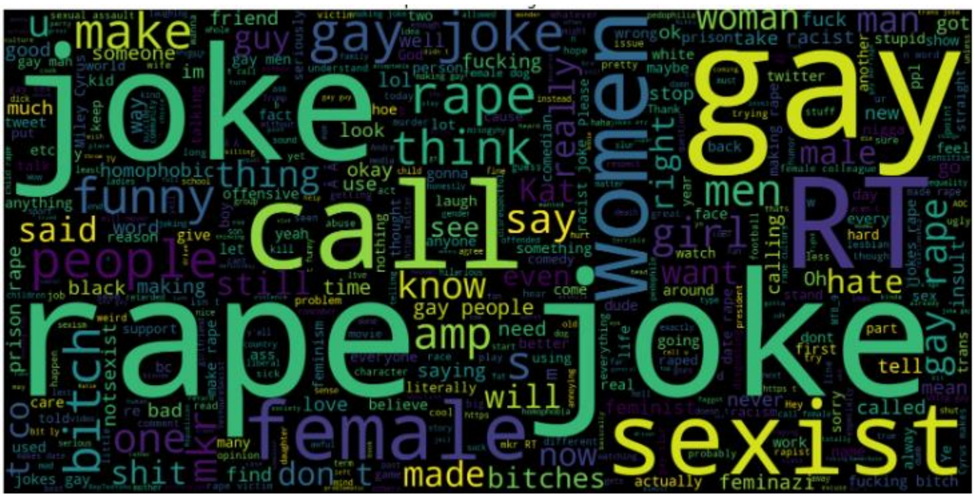


Figure 4.3.1.4 Frequent words in gender cyberbullying type

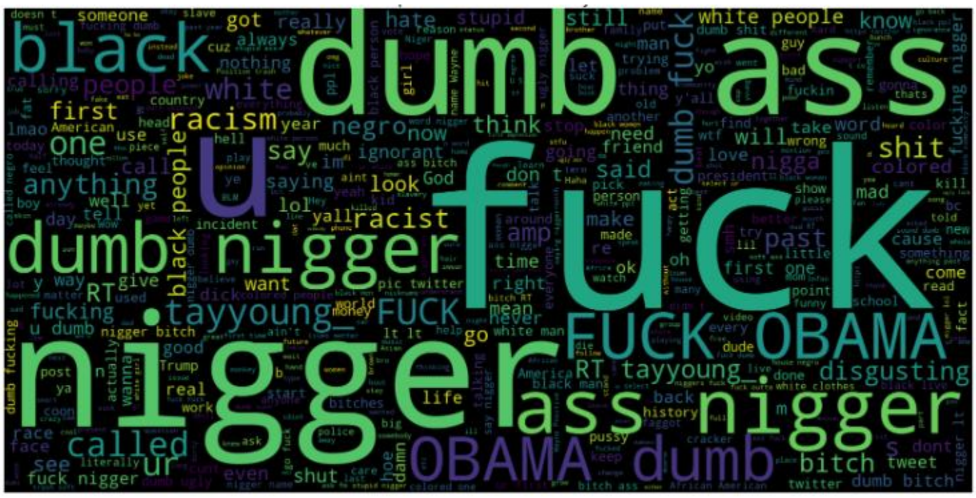


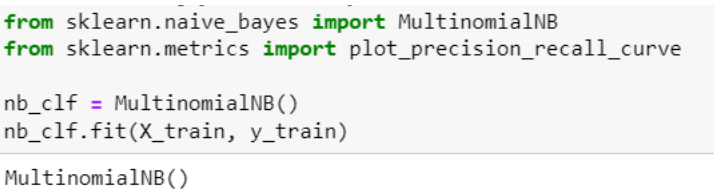
Figure 4.3.1.5 Frequent words in ethnicity cyberbullying type

4.3.2. Build model

Based on the previous step, the pre-process data will go to the next step which is the TF-IDF process. After all these, the train and tested data now will undergo the training model process to get the best accuracy for each algorithm; Naïve Bayes algorithm, Support Vector Machine (SVM) algorithm and Decision Tree algorithm.

Figure 4.3.2.1 shows the method for build the Naïve Bayes model. Under the scikit-learn library, there are three types of Naïve Bayes model. In this study, the type of Naïve Bayes that being used is the Multinomial Naïve Bayes.

The types of Naïve Bayes model are Gaussian Naïve Bayes, Multinomial Naïve Bayes and Bernoulli Naïve Bayes. The Gaussian model presupposes that characteristics are distributed normally. This indicates that the model thinks that predictor values are samples from the Gaussian distribution if they take continuous values rather than discrete ones. Multinomial Naïve Bayes is used when there is discrete count in the data. It is suitable to use when the dataset that being used have many words. This type of Naïve Bayes is commonly used when there is a text classification problem. The Bernoulli method is used when the dataset consists of binary data.

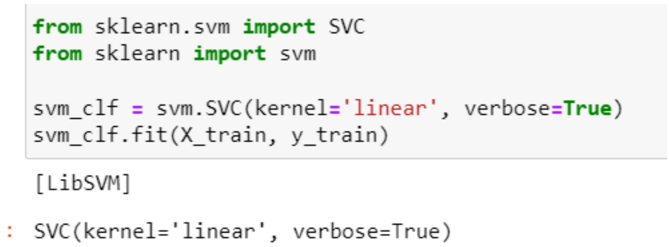


```
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import plot_precision_recall_curve

nb_clf = MultinomialNB()
nb_clf.fit(X_train, y_train)
```

```
MultinomialNB()
```

Figure 4.3.2.1 Building Naive Bayes model

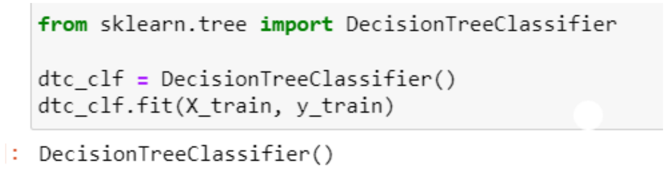
Figure 4.3.2.2 shows the method used to build the model Support Vector Machine (SVM) algorithm. The type of SVM that being used in this study is the linear SVM.



```
from sklearn.svm import SVC
from sklearn import svm

svm_clf = svm.SVC(kernel='linear', verbose=True)
svm_clf.fit(X_train, y_train)

[LibSVM]

: SVC(kernel='linear', verbose=True)
```

Figure 4.3.2.2 Building model for Support Vector Machine (SVM)

Figure 4.3.2.3 show the method that is used to build the Decision Tree model before further to the model evaluation process.



```
from sklearn.tree import DecisionTreeClassifier

dtc_clf = DecisionTreeClassifier()
dtc_clf.fit(X_train, y_train)

: DecisionTreeClassifier()
```

Figure 4.3.2.3 Building model for Decision Tree algorithm

4.3.3 Model evaluation

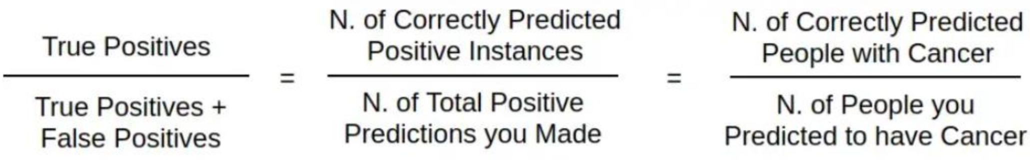
Based on previous step, three model has been built which is Naïve Bayes model, Support Vector Machine (SVM) mode, and Decision Tree model. These models are built to find the accuracy between the algorithm and compare the accuracy among them. Table 4.3.3.1 shows the summary and the comparison of the accuracy that obtained from each model. Based on the table below, the highest accuracy results that produced for detecting the cyberbullying text is from Support Vector Machine (SVM) model with 92.87% accuracy. Decision Tree model produced the second highest accuracy with 91.65% while Naïve Bayes produced the lowest accuracy among these three models which is 83.67%.

Table 4.3.1 Accuracy obtained from each model

Accuracy (%)	
Model	Accuracy obtained
Naïve Bayes	0.8367 (83.67%)
Support Vector Machine (SVM)	0.9287 (92.87%)
Decision Tree	0.9165 (91.65%)

The percentage of labels that were correctly predicted positively is represented by the model accuracy score. Another name for precision is the positive predictive value. False positives and false negatives are traded off using precision together with recall. The class distribution has an impact on precision. Precision will be worse if there are more samples in the minority class. One way to think of precision is as an indicator of accuracy or calibre. A model with high accuracy is the one we would use if we wanted to reduce false negatives. Contrarily, we would pick a model with high recall if we wanted to reduce the number of false positives.

Precision is mostly employed when we need to anticipate the positive class, and erroneous positives have a higher cost than false negatives. When the classes are severely unbalanced, the accuracy score is a helpful indicator of prediction success. It indicates the ratio of true positive to the total of true positive and false positive in mathematics. The formula to calculate precision is shown as below.



$$\frac{\text{True Positives}}{\text{True Positives + False Positives}} = \frac{\text{N. of Correctly Predicted Positive Instances}}{\text{N. of Total Positive Predictions you Made}} = \frac{\text{N. of Correctly Predicted People with Cancer}}{\text{N. of People you Predicted to have Cancer}}$$

Figure 4.3.3.1 Formula of precision [11]

Table 4.3.3.2 is the summary and comparison of the precision value that obtained from the calculation using Python programming language. Based on the results, the precision model that produced. Based on the table below, the highest precision results that produced is from Support Vector Machine (SVM) model with 93% precision. Decision Tree model produced the second highest precision with 92% while Naïve Bayes produced the lowest precision value among these three models which is 84%.

Table 4.3.2 Precision value obtained from each model

Precision (%)	
Model	Precision obtained
Naïve Bayes	84%
Support Vector Machine (SVM)	93%
Decision Tree	92%

The model's ability to properly forecast positives out of real positives is measured by the model recall score. This differs from precision, which counts the proportion of accurate positive predictions among all positive predictions given by models. The recall score would be the percentage of positive reviews that your machine learning model properly identified as positive, for instance, if you were trying to identify positive reviews. In other words, it assesses how well our machine learning model is able to distinguish between all true positives and all false positives inside a dataset. The machine learning model is more adept at recognising both positive and negative samples the higher the recall score.

Sensitivity or the true positive rate are other names for recall. A high recall score shows how well the model can locate examples of success. On the other hand, a low recall score shows that the model is poor in identifying success stories. To provide a comprehensive view of the model's performance, recall is sometimes combined with additional performance measures like precision and accuracy. It symbolises the ratio of true positives to the total of true positives and false negatives in mathematics. The formula to calculate recall is shown as below.

$$\frac{\text{True Positives}}{\text{True Positives + False Negatives}} = \frac{\text{N. of Correctly Predicted Positive Instances}}{\text{N. of Total Positive Instances in the Dataset}} = \frac{\text{N. of Correctly Predicted People with Cancer}}{\text{N. of People with Cancer in the Dataset}}$$

Figure 4.3.3.2 Formula of recall [11]

Table 4.3.3.3 is the summary and comparison of the recall value that obtained from the calculation using Python programming language. Based on the results, the precision model that produced. Based on the table below, the highest recall results that produced is from Support Vector Machine (SVM) model with 93% precision. Decision Tree model produced the second highest recall value with 92% while Naïve Bayes produced the lowest recall value among these three models which is 84%.

Table 4.3.3 Recall value obtained from each model

Recall (%)	
Model	Recall obtained
Naïve Bayes	84%
Support Vector Machine (SVM)	93%
Decision Tree	92%

The model score as a function of recall and accuracy is represented by the model F1 score. A substitute for accuracy measures (it doesn't require us to know the entire number of observations), the F-score is a machine learning model performance statistic that equally weights precision and recall when assessing how accurate the model is. It is frequently used as a single value that offers summaries of the model's output quality. This is a helpful model measurement in situations where attempting to maximise either accuracy or recall score results in decreased model performance. The formula to calculate F1-Score is shown as below.

$$2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Figure 4.3.3.3 Formula of F1-Score [11]

Table 4.3.3.4 is the summary and comparison of the F1-Score that obtained from the calculation using Python programming language. Based on the results, the precision model that produced. Based on the table below, the highest F1-Score results that produced is from Support Vector Machine (SVM) model with 93% precision. Decision Tree model produced the second highest F1-Score with 92% while Naïve Bayes produced the lowest F1-Score among these three models which is 82%.

Table 4.3.4 F1-Score obtained from each model

F1-Score (%)	
Model	F1-Score obtained
Naïve Bayes	82%
Support Vector Machine (SVM)	93%
Decision Tree	92%

4.3 CONCLUSION

From all the tables that shown in 4.2, Support Vector Machine (SVM) algorithm produce the highest value for accuracy, F1-Score, precision and recall which is 93%. Decision Tree algorithm, also produce a consistent value for each test which is 92% while the last Naïve Bayes produce 84% value at the recall and precision testing and produce 82% at F1-Score.

From this observation, this thesis can conclude that, Support Vector Machine (SVM) is the best algorithm that can be used to detect the cyberbullying text.

CHAPTER 5

5.1 INTRODUCTION

This chapter will summarize all the content of this research paper together with the objective revisited, limitation and future works.

This research is about the detection of cyberbullying using Machine Learning approach. Several algorithms from Machine Learning were proposed to develop the model and finding the accuracy between the algorithms; Support Vector Machine (SVM), Naïve Bayes and Decision Tree. There are test were performed to find the best accuracy for these three algorithms. From this approach, SVM approach will produce the best results for each test such as accuracy, precision, recall and F1-Score with 93% result. Thus, SVM is identified as the best algorithm for cyberbullying detection in this thesis.

5.2 OBJECTIVE REVISITED

As discussed in Chapter 1, there are three objective that this research needs to achieve. The first objective of this research is to study the existing technique for cyberbullying detection. This objective is achieved by reviewing the previous research work related to the cyberbullying detection. The details of this previous research paper have been discussed in Chapter 2.

The next objective is to evaluate the cyberbully detection based on the accuracy from three classifiers which are Support Vector Machine (SVM), Naïve Bayes and Decision Tree. This objective is achieved by using the Python Programming Language to produce the accuracy for each algorithm. The details of this process can be seen in Chapter 4.

The last objective for this thesis is to compare the classifiers precision output for detecting cyberbullying text. This objective is achieved by comparing all the accuracy obtained from the algorithms. Not only that, the recall, precision, and F1-Score also obtained and compared in Chapter 4.

In conclusion, this study has succeeded to achieve all the stated objectives. The implementation using Machine Learning approach for cyberbullying detection and finding the highest accuracy from three algorithm has been proven that Support Vector Machine (SVM) is the best algorithm that can be used for cyberbullying text detection.

5.3 LIMITATION

In Chapter 4, to get the accuracy for each algorithm, the Python Programming Language is used. While running the algorithm for each model, the time taken for each algorithm to be executed is difference. From experienced, SVM takes the longest time to build the model and produced the accuracy. Even though SVM produce the highest accuracy, but it takes much time to process since it uses high amount of RAM. Sometimes, while running the algorithm using the software, it will make the software lagging and crash. The disadvantage using SVM to find the accuracy is that, if the file is not autosaved and the software crashed, it will make all the codes lost.

5.4 FUTURE WORKS

For future work, there are more development of algorithms under Machine Learning that can be used to detect the cyberbullying text. Since SVM tend to takes long time to produce the output other algorithms may proposed to replace the SVM algorithm to find the best accuracy. Thus, the process for building the model and produce the accuracy much more efficient.

In addition, there is system development of cyberbullying detection to analyse the sentence. From that, it will make the cyberbullying detection become more accurate since it does not have to undergo the manual coding one by one.

5.5 CONCLUSION

From the results and discussion above, this thesis can conclude that, Support Vector Machine (SVM) algorithm produce the highest value for accuracy, F1-Score, precision and recall which is 93%. Decision Tree algorithm, also produce a consistent value for each test which is 92% while the last Naïve Bayes produce 84% value at the recall and precision testing and produce 82% at F1-Score.Support Vector Machine (SVM) is the best algorithm that can be used to detect the cyberbullying text. Even though Support Vector Machine (SVM) produce the highest result for each test but it takes much time to process. Therefore, the development of new algorithms under Machine Learning that can produce the highest accuracy and takes shorter time to process is needed.

REFERENCES

[1] A. C. Baldry, D. P. Farrington, and A. Sorrentino, "'Am I at risk of cyberbullying'? A narrative review and conceptual framework for research on risk of cyberbullying and cybervictimization: The risk and needs assessment approach," *Aggression and Violent Behavior*, vol. 23, pp. 36–51, Jul. 2015, doi: 10.1016/j.avb.2015.05.014.

[2] K. Dhungana Sainju, N. Mishra, A. Kuffour, and L. Young, "Bullying discourse on Twitter: An examination of bully-related tweets using supervised machine learning," *Computers in Human Behavior*, vol. 120, p. 106735, Jul. 2021, doi: 10.1016/j.chb.2021.106735.

[3] H. Nurrahmi and D. Nurjanah, "Indonesian Twitter Cyberbullying Detection using Text Classification and User Credibility," *2018 International Conference on Information and Communications Technology (ICOIACT)*, Mar. 2018, doi: 10.1109/icoiact.2018.8350758.

[4] M. Raihan, Md. T. Islam, P. Ghosh, Md. Mehedi. Hassan, J. H. Angon, and S. Kabiraj, "Human Behavior Analysis using Association Rule Mining Techniques," *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Jul. 2020, doi: 10.1109/icccnt49239.2020.9225662.

[5] Q. Li, "Cyberbullying in High Schools: A Study of Students' Behaviors and Beliefs about This New Phenomenon," *Journal of Aggression, Maltreatment & Trauma*, vol. 19, no. 4, pp. 372–392, May 2010, doi: 10.1080/10926771003788979.

[6] Harit Shandilya, "Decision Tree. - Harit Shandilya - Medium," *Medium*, May 05, 2020. https://medium.com/@haritshandilya198/decision-tree-49ca1df60e72.

[7] ferihalditanjung, “Twitter Cyberbullying Text Classification,” *Kaggle.com*, Feb. 2022. https://www.kaggle.com/code/ferihalditanjung/twitter-cyberbullying-text-classification/notebook#Defining-Class.

[8] B. A. Talpur and D. O’Sullivan, “Multi-Class Imbalance in Text Classification: A Feature Engineering Approach to Detect Cyberbullying in Twitter,” *Informatics*, vol. 7, no. 4, p. 52, Nov. 2020, doi: 10.3390/informatics7040052.

[9] Noviantho, S. M. Isa, and L. Ashianti, “Cyberbullying classification using text mining,” *2017 1st International Conference on Informatics and Computational Sciences (ICICoS)*, 2017. https://www.semanticscholar.org/paper/Cyberbullying-classification-using-text-mining-Noviantho-Isa/48b502c0baf2a7f1bf66deb595ec5ffa5c2f447f.

[10] M. Guia, Rodrigo Rocha Silva, and J. Bernardino, “Comparison of Naïve Bayes, Support Vector Machine, Decision Trees and Random Forest on Sentiment Analysis,” *ResearchGate*, 2019. https://www.researchgate.net/publication/336225950_Comparison_of_Naive_Bayes_Support_Vector_Machine_Decision_Trees_and_Random_Forest_on_Sentiment_Analysis.

[11] Teemu Kanstrén, “A Look at Precision, Recall, and F1-Score - Towards Data Science,” *Medium*, Sep. 11, 2020. https://towardsdatascience.com/a-look-at-precision-recall-and-f1-score-36b5fd0dd3ec#:~:text=F1%2DScore%20is%20a%20measure,than%20the%20traditional%20arithmetic%20mean

[12] R. Agrawal, “Must Known Techniques for text preprocessing in NLP,” *Analytics Vidhya*, Jun. 14, 2021. https://www.analyticsvidhya.com/blog/2021/06/must-known-techniques-for-text-preprocessing-in-nlp/

[13] "Crisp DM methodology - Smart Vision Europe," *Smart Vision Europe*, Jun. 17, 2020. https://www.sv-europe.com/crisp-dm-methodology/#three

[14] P. Sharma, "Decision Tree Classification | Guide to Decision Tree Classification," *Analytics Vidhya*, Apr. 29, 2021. https://www.analyticsvidhya.com/blog/2021/04/beginners-guide-to-decision-tree-classification-using-python/

[15] T. Joachims, "Text categorization with Support Vector Machines: Learning with many relevant features," *Machine Learning: ECML-98*, pp. 137–142, 1998, doi: 10.1007/bfb0026683.

[16] "Big Data." [Online]. Available: http://dhoto.lecturer.pens.ac.id/lecture_notes/internet_of_things/Big%20Data%20Principles%20and%20Paradigms.pdf

[17] lizakonopelko, "Cyberbullying on Twitter Visualization," *Kaggle.com*, Jan. 17, 2022. https://www.kaggle.com/code/lizakonopelko/cyberbullying-on-twitter-visualization/notebook#Age-based-hate-on-twitter

[18] S. Li, "Multi-Class Text Classification Model Comparison and Selection," *Medium*, Sep. 25, 2018. https://towardsdatascience.com/multi-class-text-classification-model-comparison-and-selection-5eb066197568

[19] "Text Cleaning for NLP: A Tutorial," *MonkeyLearn Blog*, May 31, 2021. https://monkeylearn.com/blog/text-cleaning/

[20] C. Zhu, S. Huang, R. Evans, and W. Zhang, "Cyberbullying Among Adolescents and Children: A Comprehensive Review of the Global Situation, Risk Factors, and Preventive Measures," *Frontiers in Public Health*, vol. 9, Mar. 2021, doi: 10.3389/fpubh.2021.634909.

[21] M. Islam, Md Ashraf Uddin, L. Islam, and U. K. Acharjee, "Cyberbullying Detection on Social Networks Using Machine Learning

Approaches," *ResearchGate*, Apr. 28, 2021. https://www.researchgate.net/publication/351131976_Cyberbullying_Detection_on_Social_Networks_Using_Machine_Learning_Approaches

[22] kirtiksingh, "Cyberbullying-Detection-using-Machine-Learning/Notebook.ipynb at main · kirtiksingh/Cyberbullying-Detection-using-Machine-Learning," *GitHub*, 2023. https://github.com/kirtiksingh/Cyberbullying-Detection-using-Machine-Learning/blob/main/Notebook.ipynb

[23] K. Ross, "Linear vs. Non-linear Support Vector Machines: Which is More Accurate?," *Medium*, Jun. 05, 2019. https://medium.com/@krr3wf/linear-vs-non-linear-support-vector-machines-which-is-more-accurate-d2380fcd57e6