

CYBERBULLYING DETECTION: A MACHINE LEARNING APPROACH

BY

YEONG SU YEN

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

MAY 2022

REPORT STATUS DECLARATION FORM

Title: CYBERBULLYING DETECTION: A MACHINE LEARNING APPROACH

Academic Session: MAY 2022

I YEONG SU YEN

(CAPITAL LETTER)

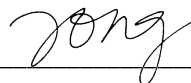
declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,



(Author's signature)



(Supervisor's signature)

Address:

10, PALMA D/1, SERI PALMA,
BANDAR SERI BOTANI,
31350, IPOH, PERAK

DR. TONG DONG LING
Supervisor's name

Date: 9 SEPTEMBER 2022

Date: 9 SEPTEMBER 2022

Universiti Tunku Abdul Rahman			
Form Title : Sample of Submission Sheet for FYP/Dissertation/Thesis			
Form Number: FM-IAD-004	Rev No.: 0	Effective Date: 21 JUNE 2011	Page No.: 1 of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 5 SEPTEMBER 2022

SUBMISSION OF FINAL YEAR PROJECT

It is hereby certified that Yeong Su Yen (ID No: 18ACB01410) has completed this final year project entitled “Cyberbullying Detection: A Machine Learning Approach” under the supervision of Dr. Tong Dong Ling (Supervisor) from the Department of Computer Science, Faculty of Information and Communication Technology.

I understand that University will upload softcopy of my final year project in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,



(*Yeong Su Yen*)

*Delete whichever not applicable

DECLARATION OF ORIGINALITY

I declare that this report entitled “**CYBERBULLYING DETECTION: A MACHINE LEARNING APPROACH**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.



Signature : _____

Name : Yeong Su Yen

Date : 9 September 2022

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Dr. Tong Dong Ling and my moderator, Dr Jasmina Khaw Yen Min, who has given me this bright opportunity to engage in a text classification and sentiment analysis project. I am also thankful that I have an opportunity to build a web application. I have learnt a lot of useful knowledge while completing this project and I will apply these skills obtained in my career. A million thanks to you.

Finally, I must say thanks to my parents and my family for their love, support, and continuous encouragement throughout the course.

ABSTRACT

Machine learning is a hot topic and it is widely implemented in software, web application and more. Those algorithms are used in the classification or regression model to predict an input. Nowadays, the cases of cyberbullying have been increasing over the years. It causes distress to those that are involved, even though they are not hurt physically but they are mentally affected. Even though the social media sites have been taking measures to control the situation, and it helped to decrease the cyberbullying cases. However, it might not be enough because not every social media site has a cyberbullying detector machine. In this project, a model was created to classify the text as cyberbullying message or non-cyberbullying message. This model combines a rule-based approach of sentiment analysis and a supervised machine learning algorithm to classify the text. This model used sentiment analysis to label the datasets and these data are fed into the classifier for training. TextBlob was used to determine the polarity of the text. After labelling the data, these labels will act as the target feature for the model. Bag of Words model was used to convert text into numerical inputs. The machine learning algorithm, Support Vector Machine was chosen after comparing it with other algorithms such as Multinomial Naïve Bayes, Decision Tree Classifier, and Random Forest Classifier. The model has a high accuracy score, 0.93. The F1-score for both classes were high, 0.92 for non-cyberbullying class, 0.93 for cyberbullying class. Finally, the model was pickled and loaded into the web application. The web application was created to test the effectiveness of the model, it would simulate the process of cyberbullying that will occur in a social media site.

TABLE OF CONTENTS

TITLE PAGE	I
REPORT STATUS DECLARATION FORM	II
DECLARATION OF ORIGINALITY	IV
ACKNOWLEDGEMENTS	V
ABSTRACT.....	VI
LIST OF FIGURES	X
LIST OF TABLES	XII
LIST OF ABBREVIATIONS	XIII
CHAPTER 1	1
INTRODUCTION.....	1
1.1 Problem Statement and Motivation	1
1.2 Objectives	2
1.3 Project Scope and Direction.....	2
1.4 Contributions.....	3
1.5 Background Information.....	4
1.6 Timeline	4
1.7 Report Organization.....	4
CHAPTER 2.....	6
LITERATURE REVIEW	6
2.1 Articles related to Machine Learning Model	6
2.1.1 Cyberbullying Detection Using Machine Learning.....	6
2.1.2 Automatic cyberbullying detection: A systematic review [4]	7

2.1.3 An Empirical Study and Analysis of the Machine Learning Algorithms Used in Detecting Cyberbullying in Social Media [5].....	8
2.1.4 Hybrid approach: naive bayes and sentiment VADER for analyzing sentiment of mobile unboxing video comments [6]	9
2.2 Web Application Review	10
2.2.1 Profanity Detector	10
CHAPTER 3 SYSTEM METHODOLOGY/APPROACH.....	11
3.1 System Design Diagram	11
3.1.1 System Architecture Diagram.....	11
3.1.2 User Requirements.....	12
3.1.3 Use Case Diagram and Description	12
CHAPTER 4 SYSTEM DESIGN	19
4.1 Machine Learning Model.....	19
4.1.1 Data Preprocessing.....	19
4.1.2 Data Labelling.....	20
4.1.3 Find the Most Suitable Machine Learning Algorithm	23
4.1.4 The Final Machine Learning Model	34
4.2 Web Application	35
4.2.1 Creating a Database (SQLite)	37
4.2.2 Wireframe of Web Pages	38
4.2.3 Building a Corpus to Store All Abusive Words.....	42
4.2.4 A Function to Determine the Likelihood of Cyberbullying.....	44

4.2.5 Final Design of all Web Pages.....	45
CHAPTER 5 SYSTEM IMPLEMENTATION	46
5.1 Hardware Setup.....	46
5.2 Software Setup	46
5.3 System Settings and Configuration.....	47
5.4 System Operation (with Screenshot)	49
CHAPTER 6 SYSTEM EVALUATION AND DISCUSSION	52
6.1 System Testing and Performance Metrics	52
6.2 Testing Setup and Result	52
6.3 Project Challenges	55
6.4 Objectives Evaluation	55
CHAPTER 7 CONCLUSION AND RECOMMENDATION.....	56
7.1 Conclusion	56
7.2 Recommendation	56
REFERENCES.....	57
APPENDICES	A-1
APPENDIX A: WEEKLY LOG.....	A-1
APPENDIX B: CODES.....	B-1

POSTER

PLAGIARISM CHECK RESULT

CHECKLIST FOR FYP2

LIST OF FIGURES

Figure	Title	Page
Figure 1.6.1	Gantt Chart for FYP2	5
Figure 2.2.1.1	Profanity Detector	9
Figure 3.1	Architecture Diagram of Web Application	10
Figure 3.2	Figure 3.2 Use Case Diagram of the Web Application	12
Figure 4.1	Figure 4.1 Text Blob	20
Figure 4.2	Figure 4.2 VADER	20
Figure 4.3	Figure 4.3 Bar Chart For TextBlob	21
Figure 4.4	Figure 4.4 Bar Chart for VADER	21
Figure 4.5	Figure 4.5 Confusion Matrix of SVM	27
Figure 4.6	Figure 4.6 Confusion Matrix of random forest	27
Figure 4.7	Figure 4.7 Classification results of Base Model	28
Figure 4.8	Figure 4.8 ROC curve of base model	29
Figure 4.9	Figure 4.9 Results after grid search	30
Figure 4.10	Figure 4.10 The results of base model with TFIDF	31
Figure 4.11	Figure 4.11 Results after grid search (TFIDF)	32
Figure 4.12	Figure 4.12 ROC curve of both fined tuned model	33
Figure 4.13	Figure 4.13 Precision Recall Graph for both models with BOW	33
Figure 4.14	Block Diagram of the Cyberbullying Classifier	34
Figure 4.15	Structure of the Web Application	35
Figure 4.16	ERD of Database	37
Figure 4.17	Wireframe of Index.html	38
Figure 4.18	Wireframe of Blog.html	39
Figure 4.19	Wireframe of Stats.html	41
Figure 4.20	Likelihood of cyberbullying cases	44
Figure 4.21	Design of the Blog Page	44
Figure 4.22	Design of the Home page	45
Figure 4.23	Design of the Statistics Page	45
Figure 5.1	Anaconda prompt	47

Figure 5.2	Jupyter notebook home page	47
Figure 5.3	The page that has the codes	48
Figure 5.4	Shutdown the notebook	48
Figure 5.5	Run web application	49
Figure 5.6	Link to web app and close web app	49
Figure 5.7	Message typed into the form	49
Figure 5.8	Error message shown	50
Figure 5.9	Create Blog Post	50
Figure 5.10	Show Blog Post created	51
Figure 5.11	Statistic page will show the updated data	51

LIST OF TABLES

Table Number	Title	Page
Table 3.1.1	Functional Requirements Listing for F002	12
Table 3.1.2	Use Case Description for [F001]	12
Table 3.1.3	Functional Requirements Listing for F002	13
Table 3.1.4	Use Case Description for [F002]	14
Table 3.1.5	Functional Requirements Listing for F004	14
Table 3.1.6	Use Case Description for [F003]	15
Table 3.1.7	Functional Requirements Listing for F005	15
Table 3.1.8	Use Case Description for [F004]	16
Table 3.1.9	Functional Requirements Listing for F006	17
Table 3.1.10	Use Case Description for [F005]	17
Table 3.1.11	Functional Requirements Listing for F007	17
Table 3.1.12	Use Case Description for [F006]	18
Table 4.1.1	Types of Dataset Used to Train the Model	24
Table 4.1.2	Results of 10 fold validation	26
Table 4.2.1	Description of Attributes	36
Table 4.2.1	The Name of the Files Generated	42
Table 5.2.1	Hardware used in the project	44
Table 5.2.2	Software used in the project	45
Table 6.2.1	Results of all Test Cases	53

LIST OF ABBREVIATIONS

API	Application Interface
ASCII	American Standard Code for Information Interchange
BOW	Bag of Words
CSV	Comma-separated values
HTTP	Hypertext Transfer Protocol
<i>HTML</i>	HyperText Markup Language
IDF	Inverse Document Frequency
KNN	K-Nearest Neighbors
MCC	Matthews Correlation Coefficient
NLP	Natural language processing
ORM	Object Relational Mapping
POS	Part-of-Speech
RAM	Random-Access Memory
RBF	Radial basis function
SMOTE	Synthetic Minority Oversampling Technique
TF-IDF	Term Frequency–Inverse Document Frequency
URL	Uniform Resource Locator
UTF-8	Unicode Transformation–8-bit
WTF	WT Forms

Chapter 1

Introduction

1.1 Problem Statement and Motivation

As technology advances, software developers had developed social media applications that allow people to communicate with each other instantly. For example, Facebook, Twitter, Instagram, Tik Tok and more. Besides that, social media applications allow people to post an image or video online. People can write a comment on the posted pictures or videos in the comment section. However, some people misused this platform to bully people online. When someone is being bullied online, it is known as cyberbullying [1]. The methods of cyberbullying are sending messages or posting posts that are abusive, mean, and hurtful to someone else. It also includes spreading rumours about someone else. It is easier to target someone online because they can hide their identity by not disclosing their names and profile pictures. Most of the victims are young adults and teenagers. This is because they are active users of social media application.

Victims are afraid to voice out when they are being harassed online. Most of the victims know that they are being bullied online but they just ignore it. Even though there are a lot of programmes that educate the public about cyberbullying, it is still important for the victims to take the initiative to stop it before it gets worse. If cyberbullying persists for a long time, it will affect the mental health of the victim and it might affect their studies resulting in poor performance, they might get depressed, or even suicidal thoughts.

There is a lack of website that can detect sentence that has abusive words and the emotion of the text. For example, Readable Pro has a profanity detector that can detect swear words in a text. A website called Noswearing is used to filter out the swear word and replace it with a word that has a similar meaning. These websites mentioned above can detect swear words only. It cannot detect the polarity of the sentence.

This project would be focusing on building a model that can classify the text into cyberbullying or non-cyberbullying. The model would be loaded into the web application. The purpose of creating a web application was to simulate the cyberbullying activities that might occur in a social media site.

1.2 Objectives

There are a lot of studies conducted about detecting text related to cyberbullying on social media platforms using text mining, machine learning algorithms, and sentiment analysis. However, most of these models are not integrated into a web application. Users that did not learn text mining might not be able to use it or interpret the results. Thus, the objective was to test if the cyberbullying classification model would work well in the web application to detect the sentence with words that has abusive, offensive, or harmful meaning.

The proposed project should correctly identify message that belongs to the cyberbullying class. The functions of finding the abusive words in the message and the likelihood of cyberbullying would act as evidence to prove that cyberbullying might have occurred or not occurred. This project also utilized the rule-based approach to find out the sentiment of the sentence [26]. This method was efficient to label the sentence because it automatically identifies their respective polarity. With the help of sentiment analysis, it could find out the emotion the text is conveying. The emotions such as sadness, happiness, or anger when someone reads it. The labels would be the target features that would be used in the model.

This project aims to find the most suitable machine learning algorithm that can correctly predict a message that has a negative meaning. Therefore, a separate program was written to find the If the polarity of the message is negative, it will be classified as a cyberbullying text.

1.3 Project Scope and Direction

This project would develop a model that can classify the message into their respective category such as cyberbullying and non-cyberbullying. It combined a rule-based approach of sentiment analysis and a machine learning algorithm to classify the text. The polarity of message would be categorized as negative and positive. Therefore, the machine learning model would be trained to learn that a negative message represents cyberbullying message. If the polarity of the sentence was detected as negative, it showed a higher chance that the user was being cyberbullied. The machine learning model would classify the negative messages as cyberbullying text and positive messages as non-cyber bullying text.

It requires 4 phases to create a model which is data pre-processing, labelling the datasets, training and testing the model. Data must be cleaned to maintain the quality of the data because

a noisy dataset will affect the performance of the model in a training phase [25]. It is important to get a high accuracy for this model to avoid misclassification. The text was labelled using a rule-based approach of sentiment analysis and these labels would act as the target feature of the model. After the data is cleaned, it can be observed that most of the preprocessed text did not alter the meaning of the original text. This is an important step to get a high accuracy for finding out the polarity of the text. Furthermore, the text is labelled using a rule-based approach of sentiment analysis. Rule-based approaches of sentiment analysis are VADER and TextBlob. After a comparison of both approaches, TextBlob is chosen to label the dataset. This is because it works well on both cleaned and preprocessed datasets. Most of the polarity of the original text and the preprocessed text remained unchanged. Then, the text is converted to numerical form. Next, the algorithm will classify the text into binary classes. The results generated by the model after training would be evaluated. The model would be fine tune to find the best parameters to build a better classifier. The model will be tested with the test set. Finally, the model would be pickled.

The model was integrated into the web application that requires the user to input sentences that contain abusive, swear, and profanity words. The web application consists of a database and the data from user would be saved in the database. The web application has 3 web pages, “Home”, “Blog” and “Statistics”. User was allowed to enter an input to classify the text on the web page known as “Home”. It would have a function to detect abusive words contained in the text and the likelihood of cyberbullying. It would display these results to the user. User was allowed to create and view the blog posts on web page known as “Blog”. A web page known as “Statistics” was specifically designed for the admin to track and monitor the cyberbullying activities. It would not be visible to the user. The admin would be able to access detailed information such as the posts that was classified as cyberbullying. The web application was to show the effectiveness of the model if it were to be used in a social media site.

1.4 Contributions

The cyberbullying classifier would be integrated into the web application to help the social media site to monitor and keep track of cyberbullying activities. The functions in the web application such as posts that was classified as cyberbullying and the results would be display in a table that could be easily understood by the end user. The content of the table could include the name of cyberbullies and the post that they posted.

1.5 Background Information

The cases of cyberbullying have been increasing lately, and this issue must be taken seriously as it will harm a person’s life. This project was proposed to raise the awareness of cyberbullying and encourage people to take action if they are being bullied online. The project combines machine learning algorithms, sentiment analysis and text mining to build a model that can detect text that is related to cyberbullying.

Text mining converts text that is not structured to investigate new insights [27] [17]. Most of the text from social media was in an unstructured format, so it is used to clean the text [17]. The application of text mining can be used on understanding the customer preferences based on their reviews, help doctors analyse the medical records and filter spam messages [17]. Natural language processing is one of the techniques used in text mining.

To detect harmful messages in social media, text classification is used to classify the messages into their categories. Machine learning techniques must be used to correctly predict whether the message is positive or negative.

1.6 Timeline

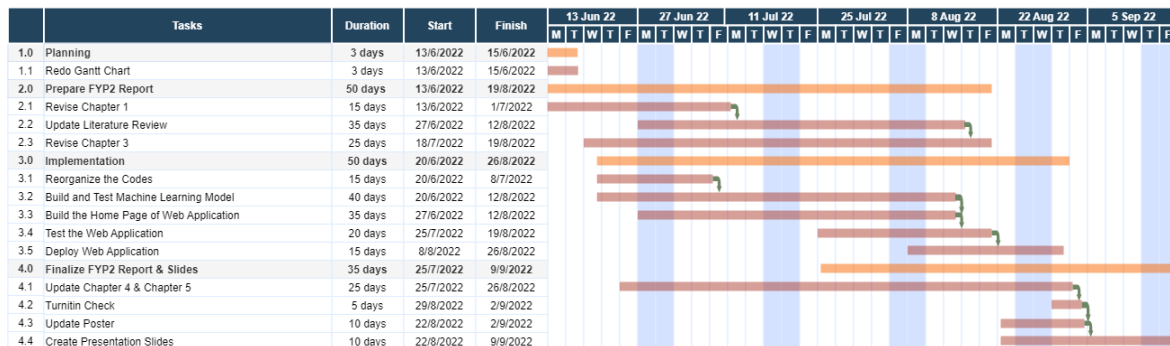


Figure 2.1.1.1 Gantt Chart FYP2

1.7 Report Organization

This report is organized into 6 chapters: Chapter 1 Introduction, Chapter 2 Literature Review, Chapter 3 System Methodology, Chapter 4 System Design, Chapter 5 System Implementation, Chapter 6 System Evaluation and Discussion, Chapter 7 Conclusion. The first chapter is the introduction of this project which includes problem statement and motivation, project background, project scope, project objectives, project contribution, timeline and report organization. The second chapter is the literature review carried out on different machine

learning model and review of a web application. The third chapter is discussing the overall system design of this project that includes the architecture diagrams and use case diagrams. The fourth chapter is about the design of the machine learning model and web application. The analysis of results would be discussed in this chapter. The fifth chapter will discuss about the system implementation. A description on how to use the programs was explained. The second last chapter would be system testing. The test would be done on the system and results would be recorded. The last chapter is the conclusion and recommendation. The appendices would be the weekly logs and codes. The poster and plagiarism results would be attached in this report.

Chapter 2

Literature Review

2.1 Articles related to Machine Learning Model

2.1.1 Cyberbullying Detection Using Machine Learning

In this study, they proposed a cyberbullying detection for messages and chats on the Discord platform to identify cyberbullying text and actors [3]. They also included a chatbot to alert bullies about the repercussion of their cyberbullying messages and will automatically report it to the appropriate channel. This study used a different approach to improve the accuracy of the classifier. They will reuse the input from the users to train the classifier.

Before using the data to train the model, they pre-processed it and applied feature extraction. They used TextBlob to determine the polarity of the data. The Naive Bayes Classifier is used to predict the labels for the message [3].

They created a function to find the vulgar confidence and created a database to store a list of swear words. If a word is found in the swear list, the vulgar confidence is returned as -1 when there are no bad words and 1 when there are. The results of the Naive Bayes Classifier model, Sentiment Analysis, and Vulgar confidence are then passed to a Combined classifier, which combines the results of the other three classifiers and returns the appropriate Cyberbullying confidence level, which is then displayed by a bot [3].

The strength of this proposed approach is they reuse the input from the users. Users with a specific role can manually report cyberbullying messages that aren't classified correctly and give them a thumbs-up reaction. It will also remove the message and the classifier will be trained with it [3]. The message is predicted with a higher cyberbullying confidence level when the user sends the same message again. When a non-cyber bullying message is misclassified, the user can give a thumbs down reaction and the classifier will be trained with this message to make sure it will be detected as positive if it is sent again.

The weakness of this project is they did not try different types of machine learning algorithms. They only used Naive Bayes algorithm. Although they can accurately classify the message, the results might improve if they have tried using other machine learning algorithms [28].

2.1.2 Automatic cyberbullying detection: A systematic review [4]

In this review, Rosa et al. analysed 22 articles related to automatic cyberbullying detection. They stated that textual features are the most common approach for feature engineering and it is often used by other researchers [4]. They also created their model after they reviewed all articles. They focused on finding different features for their model to improve the accuracy of detecting cyberbullying. They used several features such as word embeddings, sentiment features, textual features, personality trait features, and MRC psycholinguistic features for their model [4]. They combined each feature with tfidf to represent different settings to test it on the model [29].

In this study, they used two different datasets for testing and several classifiers [4]. The datasets were tested using a machine learning classifier with different scenarios to get different results.

The strength of this study is they extracted different features and used different machine learning classifiers to find the model that has the best performance. However, the classifiers did not perform well for the second dataset. Even though they extracted various features to improve the models but the results were not good because of the low f1-score and recall. They stated that sentiment analysis is used but it is a complicated task [4]. They stated that only 3 articles used it because may cause an error during classification. The sentiment features were trained with SVM and the results of the model improved when compared with the other 2 classifiers. However, it is not the best performance when compared with other features.

2.1.3 An Empirical Study and Analysis of the Machine Learning Algorithms Used in Detecting Cyberbullying in Social Media [5]

In this article, M.Sintaha and M.Mostakim introduced a system to detect cybercrime [5]. The datasets are obtained using Twitter Application Interface (API). They corrected the spelling mistakes, changed text into small capital letters, and dropped stop words from the dataset [5]. They built a list of the emoticons in utf-8 format to improve the data-preprocessing [5]. They replaced the emoji with a name that describes the emoji [24]. This is because users commonly insert emojis into their text messages to express their emotions. This will help to determine the polarity of the sentence.

In this article, they applied various feature extraction techniques like tokenization, punctuation, repeating letters, stop words [5]. The results showed that Support Vector Machine (SVM) has higher accuracy in predicting sentiment than Naïve Bayes.

They tried to adjust the parameters of each SVM to achieve the best results. The flaw is that they included emojis and converted it to a text format. If the sentence contains more positive words, this will help to increase its polarity. However, the inclusion of happy emoji by some cyberbullies when mocking a person may have an impact on the results.

2.1.4 Hybrid approach: naive bayes and sentiment VADER for analyzing sentiment of mobile unboxing video comments [6]

In this journal, the author combined VADER and Naive Bayes to predict sentiments of the YouTube comments [6]. This model will classify the comments into 2 categories which are negative and positive.

She used VADER to label the data into their respective categories before preprocessing the data. VADER can determine the overall polarity of the comment with a compound score. The author adjusted the range of compound scores for neutral words to decrease the number of sentences being labelled as neutral. This will increase the number of sentences that are labelled positive and negative. The adjusted range to label neutral sentence is from 0.2 and -0.2 [6]. After labelling the data, the comments that are labelled as neutral were removed.

Synthetic Minority Oversampling Technique (SMOTE) was implemented because the labelled dataset has more positive comments. It was used to balance the data by adding more samples of negative comments. The results show the classifier achieved a 79.78% accuracy and an F1 score of 83.72% [6].

The strengths of this article are the accuracy of the model is improved using the hybrid approach. It also increases the number of positives and negatives by changing range for compound scores for neutral comments. The weakness of this article is only one machine learning model was used to train the model.

2.2 Web Application Review

2.2.1 Profanity Detector

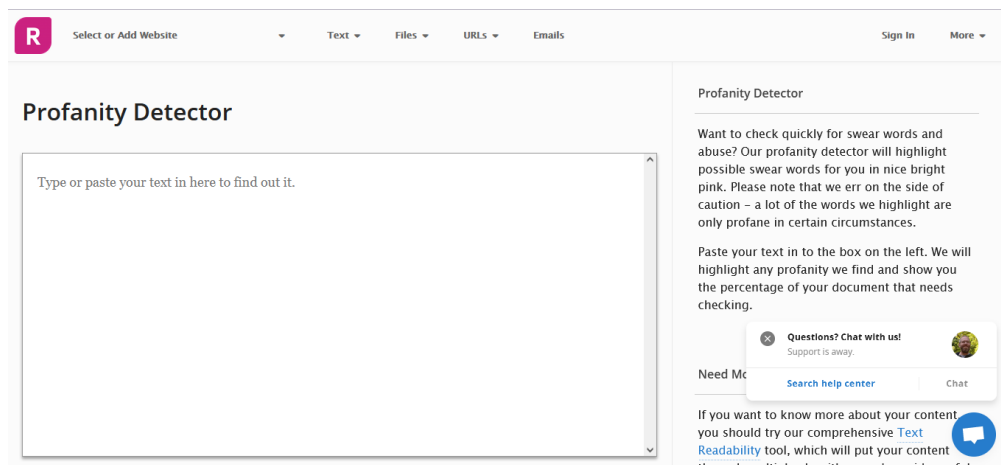


Figure 2.2.1.1 Profanity Detector

This is a website that can detect swear words or abusive words in a sentence. It allows users to directly paste the text into the text box provided by the website. It also has other analysing tools to analyse the texts such as checking the readability scores of the text, summarizing the text and more. It has no word limits. When it is scanning the text, it will show the percentage of how much text is scanned. It will calculate the sum of profanity phrase. At the same time, it will also calculate and underline the number of profanity or swear words found in the text. The profanity words will be underlined (dotted line) in a bright pink colour.

The limitation of this website is it has an API that can allow users to copy the address of the web page to scan the text but this is not applicable for profanity detection. This is the link to the website: <https://app.readable.com/text/profanity/> [19].

Chapter 3 System Methodology/Approach

3.1 System Design Diagram

3.1.1 System Architecture Diagram

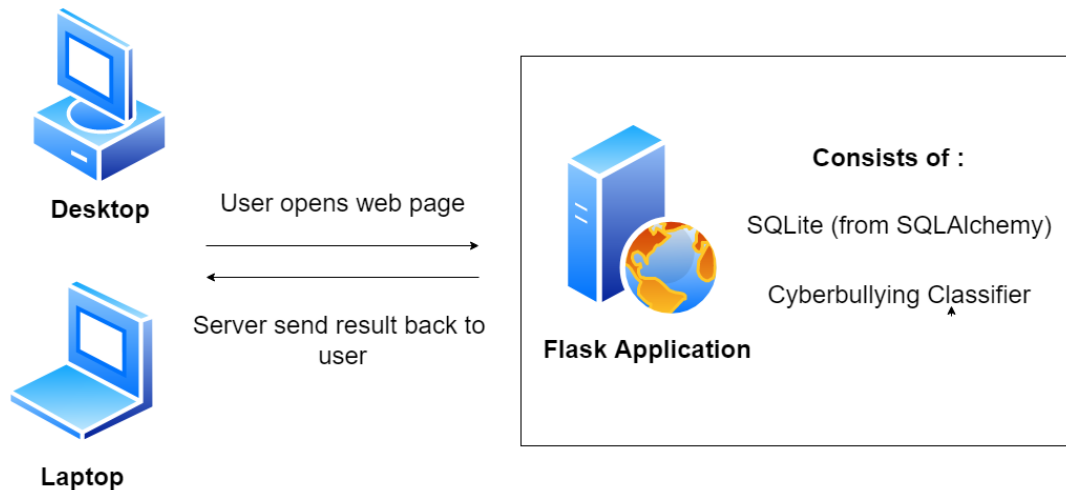


Figure 3.1.1.1 Architecture Diagram of Web Application

The architecture of the system was Client-Server model. It was chosen because all the data would store together in the same site. It was easier to manage the data. It was formed by 1 Server and 1 Client. The server was the web server. The protocol of the web server was Hypertext Transfer Protocol (HTTP) to process the requests from the client. The methods used were “GET” and “POST”.

The web application would act as the user interface. The web application was written in HTML. It has 3 web pages in the web application. Two of the three web pages would allow the client to submit a form. The first web page allows users to classify the message. After submission of the form, a request was sent to web server, Flask. The input from the form would be passed into the cyberbullying classifier for classification and some python functions. The results from the classifier and python functions would return as an output to the client, the output would be displayed to the user.

The second web page allows user to create a blog post. The input from the form will also be passed into the cyberbullying classifier and some python functions. After obtaining the results, all data would be stored into the database, SQLite that was created using SQLAlchemy. For

the third web page, it would be only accessed by the admin. All the data from the second web page would be displayed here.

3.1.2 User Requirements

1. User shall input text.
2. User shall submit form to classify message.
3. User shall view the classification results after the model classified the message.
4. User shall create blog post.
5. User shall view blog post.
6. Admin shall input text.
7. Admin shall submit form to classify message.
8. Admin shall view the classification results after the model classified the message.
9. Admin shall create blog post.
10. Admin shall view blog post.
11. Admin shall view blog activities.

3.1.3 Use Case Diagram and Description

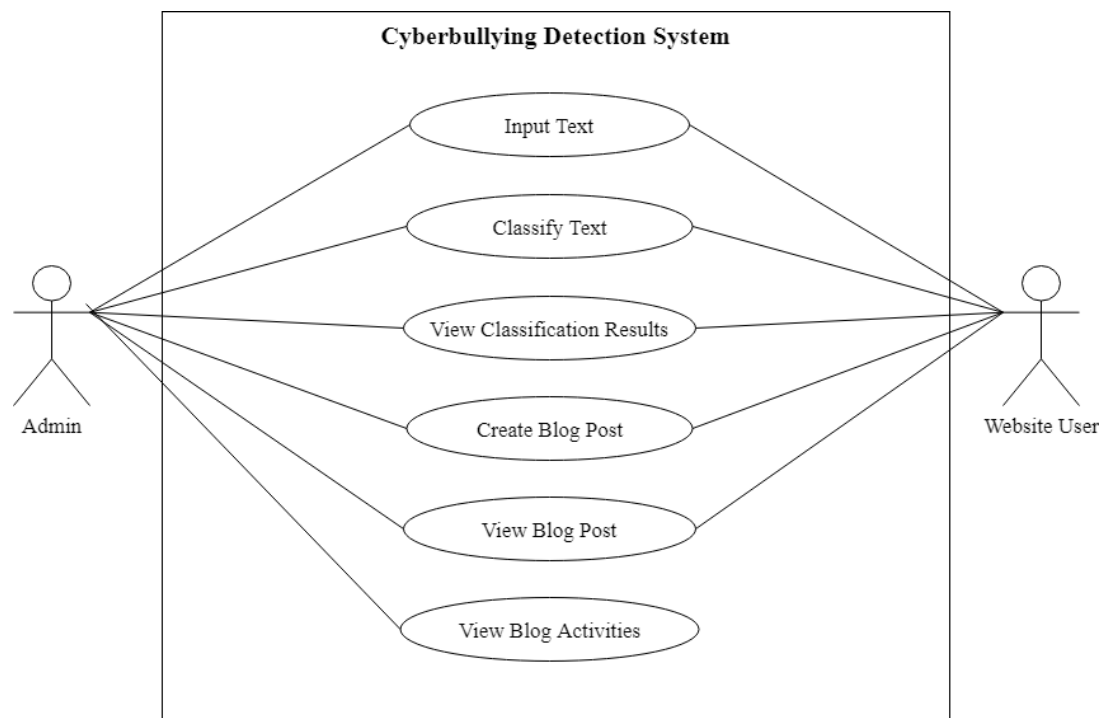


Figure 3.1.3.1 Use Case Diagram of the Web Application

3.1.3.1 [F001] Input Text

Functional Requirement ID	Functional Requirement Description
REQ_F101	System should request user to input a message.
REQ_F102	System should request user to submit form.
REQ_F103	System should allow user to input a message.
REQ_F103	System should allow user to submit form.

Table 3.1.1 Functional Requirements Listing for F001

3.1.3.2 Use Case Description for [F001]

Use Case ID	UC001	Version	1.0
Use Case	Input Text		
Purpose	Write a message that would be passed into the system for classification.		
Actor	Admin or website user		
Trigger	Admin or website user launch the system.		
Precondition	None		
Scenario Name	Step	Action	
Main Flow	1	Admin or website user launches the system.	
	2	System request admin or website user to input a message in the form.	
	3	Admin or website user input message in the form.	
	4	System request admin or website user to submit form.	
	5	Admin or website user submit form.	
Alternate Flow – Website user or Admin use Navigation Bar	1.1	Admin or website user would click on the “Home” button on the web page.	
	1.2	Back to Main Flow Step 2.	
Alternate Flow – Website user or Admin submits empty form	5.1	System will ask admin or website user to fill up the form.	
	5.2	Back to Main Flow Step 2.	
Rules	-		

Author	Yeong Su Yen
---------------	--------------

Table 3.1.2 Use Case Description for [F001]

3.1.3.3 [F002] Classify Text

Functional Requirement ID	Functional Requirement Description
REQ_F201	System should allow user to submit form.
REQ_F202	System should display result to the user.
REQ_F203	System should classify the text.
REQ_F204	System should call the function to find abusive words in the sentence.
REQ_F205	System should call the function to determine the likelihood of the cyberbullying to the user.

Table 3.1.3 Functional Requirements Listing for F002

3.1.3.4 Use Case Description for [F002]

Use Case ID	UC002	Version	1.0
Use Case	Classify Text		
Purpose	System will classify the text, find abusive word, determine likelihood of cyberbullying and generate output.		
Actor	Admin or website user		
Trigger	Admin or website user click on the “Classify Text” button.		
Precondition	After admin or website user input the message.		
Scenario Name	Step	Action	
Main Flow	1	Admin or website user click on the “Classify Text” button.	
	2	System classifies the text.	
	3	System calls the function to find abusive words in the sentence.	
	4	System calls the function to determine the likelihood of the cyberbullying to the admin or website user.	
	5	System displays the results to the admin or website user.	
Alternate Flow –	1.1	System will ask admin or website user to fill up the form.	
	1.2	Back to Main Flow Step 2.	

Website user or Admin submits empty form		
Rules	-	
Author	Yeong Su Yen	

Table 3.1.4 Use Case Description for [F002]

3.1.3.5 [F003] View Classification Results

Functional Requirement ID	Functional Requirement Description
REQ_F301	System should display abusive words found in the sentence, classification of the sentence and the likelihood of the cyberbullying to the user.
REQ_F302	System should prompt “None” if abusive word is not found in the sentence.
REQ_F303	System should clear the form and allow user to resubmit the form after viewing the results.
REQ_F304	System should process the request from user.
REQ_F305	System should allow user submits the form.
REQ_F306	System should allow user to view the classification result, abusive word found and the likelihood of the cyberbullying.

Table 3.1.5 Functional Requirements Listing for F003

3.1.3.6 Use Case Description for [F003]

Use Case ID	UC003	Version	1.0
Use Case	View Classification Result		
Purpose	System would show the results to the admin or website user.		
Actor	Admin or website user		
Trigger	After admin or website user submits the form.		
Precondition	Admin or website user click on “Classify Text” button.		
Scenario Name	Step	Action	
Main Flow	1	Admin or website user submits the form.	

	2	System processes the request from admin or website user.
	3	System displays abusive words found in the sentence, classification of the sentence and the likelihood of the cyberbullying to the admin or website user.
	4	Admin or website user view the classification result, abusive word found and the likelihood of the cyberbullying.
	5	System clears the form and allows admin or website user to resubmit the form after viewing the results.
Alternate Flow – Website user or Admin submits empty form	1.1	System will ask admin or website user to fill up the form.
	1.2	Back to Main Flow Step 2.
Alternate Flow – No abusive word found	3.1	System prompts “None” if abusive word is not found in the sentence.
	3.2	System displays the classification result and the likelihood of the cyberbullying
	3.3	Back to Main Flow Step 5.
Rules	-	
Author	Yeong Su Yen	

Table 3.1.6 Use Case Description for [F003]

3.1.3.7 [F004] Create Blog Post

Functional Requirement ID	Functional Requirement Description
REQ_F401	System should allow user to input username and message.
REQ_F402	System should allow user to submit the form.
REQ_F403	System should not allow the user to submit empty form.
REQ_F404	System should save the username and message in the database.
REQ_F405	System should classify the sentence.
REQ_F406	System should call the function to find abusive words in the sentence.
REQ_F407	System should call the function to determine the likelihood of the cyberbullying to the user.
REQ_F408	System should save the abusive words in the database.
REQ_F409	System should save likelihood of the cyberbullying to the user.

REQ_F410	System should clear the form and allow admin or website user to resubmit the form.
----------	--

Table 3.1.7 Functional Requirements Listing for F004

3.1.3.8 Use Case Description for [F004]

Use Case ID	UC004	Version	1.0
Use Case	Create Blog Post		
Purpose	System would allow admin or website user to create a blog post.		
Actor	Admin or website user		
Trigger	After admin or website user submits the form.		
Precondition	Admin or website user click on “Blog” button.		
Scenario Name	Step	Action	
Main Flow	1	System requests admin or website user to input username and message.	
	2	Admin or website user inputs username and message.	
	3	System should request admin or website user to submit the form.	
	4	System saves the username and message in the database.	
	5	System classifies the sentence.	
	6	System calls the function to find abusive words in the sentence.	
	7	System calls the function to determine the likelihood of the cyberbullying to the admin or website user.	
	8	System saves the abusive words in the database.	
	9	System saves likelihood of the cyberbullying to the admin or website user.	
	10	System will clear the form and allow admin or website user to resubmit the form.	
Alternate Flow – Website user or Admin submits empty form	3.1	System will ask admin or website user to fill up the form.	
	3.2	Back to Main Flow Step 1.	
Rules	-		
Author	Yeong Su Yen		

Table 3.1.8 Use Case Description for [F004]

3.1.3.9 [F005] View Blog Post

Functional Requirement ID	Functional Requirement Description
REQ_F501	System should display username and content of the post to the user.
REQ_F502	System should allow the user to view blog post after submitting the form.

Table 3.1.9 Functional Requirements Listing for F005

3.1.3.10 Use Case Description for [F005]

Use Case ID	UC005	Version	1.0
Use Case	View Blog Post		
Purpose	System would allow admin or website user to view a blog post.		
Actor	Admin or website user		
Trigger	After admin or website user create a blog post.		
Precondition	Admin or website user click on “Blog” button.		
Scenario Name	Step	Action	
Main Flow	1	System allows the user to view blog post after submitting the form.	
	2	System displays username and content of the post to the user.	
Rules	-		
Author	Yeong Su Yen		

Table 3.1.10 Use Case Description for [F005]

3.1.3.11 [F006] View Blog Activities

Functional Requirement ID	Functional Requirement Description
REQ_F601	System should display a table that has posts that are classified as cyberbullying, the username that posted the post and likelihood of cyberbullying based on the post.
REQ_F602	System should retrieve all the data from the database.
REQ_F603	System should display table that consist of abusive words.
REQ_F603	System should allow the user to view the tables generated.

Table 3.1.11 Functional Requirements Listing for F006

3.1.3.12 Use Case Description for [F006]

Use Case ID	UC006	Version	1.0
Use Case	View Blog Activities		
Purpose	System would allow admin or website user to view blog activities.		
Actor	Admin		
Trigger	When admin type the URL of the page “http://127.0.0.1:5000/stats”		
Precondition	None		
Scenario Name	Step	Action	
Main Flow	1	System retrieves all the data from the database.	
	2	System displays a table that has posts that are classified as cyberbullying, the username that posted the post and likelihood of cyberbullying based on the post.	
	3	System displays table that consist of abusive words.	
	4	Admin views the generated result in the tables.	
Rules	-		
Author	Yeong Su Yen		

Table 3.1.12 Use Case Description for [F006]

Chapter 4 System Design

4.1 Machine Learning Model

In this project, there were four separate programs written. The first program was the data pre-processing program. This program was done in previous semester. It was called as “Data Preprocessing- Ver 2”. The second program was to clean the table that was created in the first program and change the value of the target feature to binary numbers. The third program was created to determine the machine learning algorithm that has the highest accuracy and highest F1-score. The second program was known as “Choose the best model (final model)”. The last program was the main program as the final machine learning model. In this program, Support Vector Machine and Bag of Words model was implemented to create a cyberbullying classifier.

4.1.1 Data Preprocessing

There are a few methods to pre-process the data. Words that contain uppercase letters were converted into lowercase. Repeated words, extra spaces, usernames, links, and punctuations

were removed from the datasets. The words that are non-ASCII (American Standard Code for Information Interchange) and non-UTF8 (UCS (Unicode) Transformation Format) were removed. Tokenization was also applied to pre-process the data. It will separate the text into individual terms. The data type was string and all the words were joined together. Besides that, stop words were removed from the datasets. The stop words are words that are commonly used such as “a”, “the” and more.

Next, text normalization was used in this phase. The following methods will find the root of the word. Stemming is a crude heuristic process to trim end word [9]. It can process data faster than lemmatization. However, it might create words that do not have meaning. Lemmatization is a customized approach to determining a word's stem that employs rules based on the word's part-of-speech (POS) family [9]. The plural form of the word may not be changed to the singular form by lemmatization. Lemmatization with a Part of Speech tag was proposed to solve this problem. A package from NLTK called `averaged_perceptron_tagger` was used to find the appropriate tag for the tokens [10]. The process of this method was tagging each token according to its type and performing lemmatization after tagging it.

The comparison of these methods was saved into a csv file to look at the results in detail. After much consideration, lemmatization with pos tag was chosen as the text normalization method for this dataset to find out the root of the word. The lemmatization method was improved by labelling the words with a part of speech tag. Part of speech are nouns, adjectives, prepositions and more. After knowing the word's POS, it can correctly identify the root of the word.

4.1.2 Data Labelling

VADER and TextBlob were used. Both methods were compared to find the most suitable method to label the text. TextBlob was used to find the polarity of the data. The score that is greater than zero was labelled as “Positive”. The score that is lesser than zero is tagged as “Negative”. The score that equals zero will be tagged as “Neutral”. VADER was also used to find the polarity of the data. The rating greater or same as zero point zero five was labelled as “Positive”. The rating lesser or same as negative zero point zero five was labelled as “Negative”. The rating that is in the range from negative zero point zero five to zero point zero five will be tagged as “Neutral”.

original_tweet	cleaned_tweet	Tf	Text	TextBlob_label_original	TextBlob_label_clear
23615 @TURBOCUNT I haven't found any i'm in love with yet. i used to have some amazing pairs 10 y	haven't find im love yet use amazing pair 10 year ago	0.6	0.55	Positive	Positive
23616 Today is the day.	today day	0	0	Neutral	Neutral
23617 At least I know I'll probably never encounter him in my career, unless it's from people that ar	least know ill probably never encounter career unless people	-0.3	-0.4	Negative	Negative
23618 RT @deathomosexual: beat my ass if i ever let a man make me look stupid	beat as ever let man make look stupid	-0.8	-0.8	Negative	Negative
23619 @ZlatanDrinkin @MrStephenHowson Retard alert!!! ÁÁÁÁÁ "Á,Á"ÁÁÁ "Á,	retard alert	-1	-0.9	Negative	Negative
23620 Dude just yelled at me loud enough to hear over my music + noise canceling headphones. "Hi	dude yell loud enough hear music noise cancel headphone he	-0.1	-0.1	Negative	Negative
23621 #MKR Is honestly so fucking staged. The most over rated show after #ImACelebrityAU #MKR2I	honestly fuck stag rat show imacelebrityau mkr2015	-0	-0.4	Negative	Negative
23622 http://t.co/ZxbZV39jr: Our fluffy cat loves bottle caps and #sticks - Nala compilation http://t.	fluffy cat love bottle cap stick nala compilation coon maine n	-0.2	0.15	Negative	Positive
23623 @drurbanski we've got a PR firm. I might toss this their way.	weve get pr firm might toss way	-0.2	-0.2	Negative	Negative
23624 @classygabb hahaha you stupid! ðŸ”-	hahaha stupid	-0.4	-0.3	Negative	Negative
23625 I wanna have a FemiNazi night ðŸ”-	wan na feminazi night	0	-0.2	Neutral	Negative
23626 me: yes, bc expecting gender equality is the same as genocide	yes bc expect gender equality genocide	0	0	Neutral	Neutral
23627 @srhbutts @snipeyhead who doesn't love hugs? hugs are one of my favorite things.	doesnt love hug hug one favorite thing	0.5	0.5	Positive	Positive
23628 @rikumemes Your work is fucking terrible tbh fuck them	work fuck terrible tbh fuck	-0.7	-0.6	Negative	Negative
23629 @shereeny @caulkthewagon instead, she got a block. Because I'm used to having people shi	instead get block im use people shit mention thats shame	0	-0.2	Neutral	Negative
23630 Why do dudes have to play the "you're not serious, just wanting to fight" card when women	c dude play youre not serious want fight card woman call soluti	-0.2	-0.167	Negative	Negative
23631 @marvelousmusing I should look and see if there's anything else to add to my CV beside fem	look see there anything else add cv beside feminist attack ma	0	0	Neutral	Neutral
23632 @Shiyng_ only know how to torture & bully me	know torture bully	0	0	Neutral	Neutral
23633 @MoerasGrizzlyzly it has ties to other unpublished projects. need to publish them all at once.	tie unpublished project need publish	-0.1	0	Negative	Neutral

Figure 4.1.2.1 Text Blob

original_tweet	cleaned_tweet	Vad	Vadf	Vader_label_original	Vader_label_cleaner
23615 @TURBOCUNT I haven't found any i'm in love with yet. i used to have some amazing pairs 10 y	haven't find im love yet use amazing pair 10 year ago	0.84	0.1109	Positive	Positive
23616 Today is the day.	today day	0	0	Neutral	Neutral
23617 At least I know I'll probably never encounter him in my career, unless it's from people that are	least know ill probably never encounter career unless people	0.44	-0.052	Positive	Negative
23618 RT @deathomosexual: beat my ass if i ever let a man make me look stupid	beat as ever let man make look stupid	-0.78	-0.527	Negative	Negative
23619 @ZlatanDrinkin @MrStephenHowson Retard alert!!! ÁÁÁÁÁ "Á,Á"ÁÁÁ "Á,	retard alert	-0.47	-0.296	Negative	Negative
23620 Dude just yelled at me loud enough to hear over my music + noise canceling headphones. "HE	dude yell loud enough hear music noise cancel headphone he	0.234	-0.202	Positive	Negative
23621 #MKR Is honestly so fucking staged. The most over rated show after #ImACelebrityAU #MKR2I	honestly fuck stag rat show imacelebrityau mkr2015	0.459	-0.128	Positive	Negative
23622 http://t.co/ZxbZV39jr: Our fluffy cat loves bottle caps and #sticks - Nala compilation http://t.	fluffy cat love bottle cap stick nala compilation coon maine n	0.572	0.6369	Positive	Positive
23623 @drurbanski we've got a PR firm. I might toss this their way.	weve get pr firm might toss way	0	0	Neutral	Neutral
23624 @classygabb hahaha you stupid! ðŸ”-	hahaha stupid	-0.49	0.0516	Negative	Positive
23625 I wanna have a FemiNazi night ðŸ”-	wan na feminazi night	-0.34	0	Negative	Neutral
23626 me: yes, bc expecting gender equality is the same as genocide	yes bc expect gender equality genocide	0.402	0.4019	Positive	Positive
23627 @srhbutts @snipeyhead who doesn't love hugs? hugs are one of my favorite things.	doesnt love hug hug one favorite thing	-0.68	-0.668	Negative	Negative
23628 @rikumemes Your work is fucking terrible tbh fuck them	work fuck terrible tbh fuck	-0.8	-0.878	Negative	Negative
23629 @shereeny @caulkthewagon instead, she got a block. Because I'm used to having people shi	instead get block im use people shit mention thats shame	-0.83	-0.863	Negative	Negative
23630 Why do dudes have to play the "you're not serious, just wanting to fight" card when women	c dude play youre not serious want fight card woman call soluti	-0.27	0.4738	Negative	Positive
23631 @marvelousmusing I should look and see if there's anything else to add to my CV beside fem	look see there anything else add cv beside feminist attack ma	-0.48	-0.477	Negative	Negative
23632 @Shiyng_ only know how to torture & bully me	know torture bully	-0.8	-0.796	Negative	Negative
23633 @MoerasGrizzlyzly it has ties to other unpublished projects. need to publish them all at once.	tie unpublished project need publish	0	0	Neutral	Neutral

Figure 4.1.2.2 VADER

VADER was better at detecting abusive words accurately after analyzing and comparing both diagrams above. It can also identify offensive words. For example, line 23632 has the words “bully” and “torture”. These were negative words and it can correctly classify them as “Negative”. Using this same line, TextBlob classified it as Neutral when it contains negative words. VADER can also identify words that are typically used to express their emotion while texting someone on social media. In line 23624, it has the words “hahaha” and “stupid”. It knows that this text is a joke and classified it as Positive. TextBlob classified it as Negative because of the word “stupid”. Both methods have their strengths. Thus, a bar chart was plotted to visualize the distribution of the data.

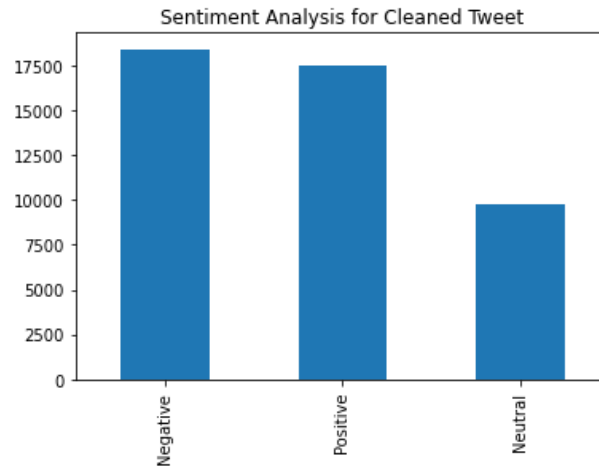


Figure 4.1.2.3 Bar Chart For TextBlob

Distribution of negative cleaned tweets and positive cleaned tweets is balanced when TextBlob was used. Most of the tweets are classified as negative.

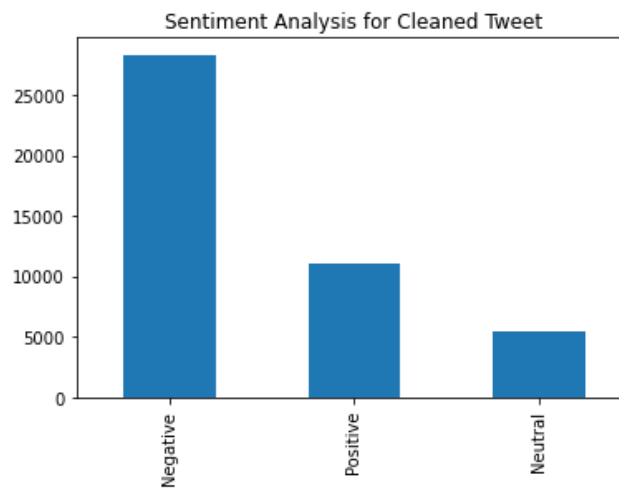


Figure 4.1.2.4 Bar Chart for VADER

The distribution of the negative cleaned tweets and positive cleaned tweets is not balanced. It shows that most of the data are labelled as negative. This dataset is a cyberbullying dataset so it will contain a lot of abusive words. The unbalanced dataset might overfit the model. In a nutshell, TextBlob was chosen to label the dataset as it works well on both cleaned and preprocessed datasets. It has a balanced class distribution.

After the data labelling phase, the tweets that are labelled as ‘Neutral’ will be dropped. Therefore, the table will only have two categories which are ‘Negative’ and ‘Positive’. This updated table will be exported as csv file named as “removedNeutralLabel.csv” to check if all “Neutral” labels are removed from the table. Some of the columns would be removed as there were many columns generated during the data pre-processing phase. The columns that contain the polarity score of the text will be renamed to “polarity”. The column that contains the polarity of the sentence would be renamed to “label”. A new column “target” will be created to use as the target feature in training the model. If the label was “Negative”, the target value was labelled as “1”. Otherwise, it was labelled as “0”. A bar chart was plotted to visualize the samples in the classes. Since the difference between the two classes were 5.32%, so the distribution of target classes was balanced. The updated table will be exported as csv file named as “cleaned_table” because it was used as the input in another separate program to obtain the most suitable machine learning algorithm.

4.1.3 Find the Most Suitable Machine Learning Algorithm

Another program was written find the most suitable algorithm to build a good classifier. This is to achieve one of the objectives in this project. A CSV file named “cleaned_table” that was created previously was loaded. The column “cleaned_tweet” was labelled as “x”, while the column “target” was labelled as “y”. The data “x” is transformed to numerical values.

This first method is the Bag of Words model. This model will create a matrix and assign the words into a separate column while each row corresponds to the list of words that was pre-processed. Other than that, term frequency-inverse document frequency can convert words to vectors or matrices because the computer cannot read words. This method might be better than Bag of Words because it can look for important words in a text. This was useful because all unique words from the sentences will be grouped into a new document. It would search for words that frequently appear in the current document to obtain a Term Frequency (TF) for each word. This is the formula of TF. i represents a term or word. d represents a document [13].

$$\text{Term Frequency } (i, d) = \frac{\text{number of times } i \text{ appears in } d}{\text{total number of terms in } d}$$

After this step, it will calculate the weightage of each word based on the total appearance to obtain the inverse document frequency. IDF will determine the importance of the word across the document. This is the formula of IDF, N represents the sum of all sentences, and df is sum of all sentences with the word [13].

After this step, it will calculate the weightage of each word based on the total appearance to obtain the inverse document frequency. IDF will determine the importance of the word across the document. This is the formula of IDF, N represents the sum of all sentences, and df is sum of all sentences with the word [13].

$$\text{Inverse Document Frequency } (i) = \log \frac{N}{1 + df}$$

TF is multiplied by IDF to get TF-IDF [13].

$$\text{TF-IDF } (i, d) = TF(i, d) \times IDF(i)$$

The words that were rare in the document will have a higher TF-IDF value. This method will help to search for important words that are related to cyberbullying across the dataset. These methods will be evaluated and compared to see which performs better.

4.1.3.1 Splitting the data into training and testing

The data loaded from the CSV file will be split into “x_partial”, “y_partial”, “x” and “y”. The total columns of “x” and “y” are both 34848. The number of columns were reduced after removing the data that was labelled as “Neutral”. The total columns of “x_partial” and “y_partial” are both 17424. It represents 50% of the complete dataset. The reason of creating this dataset was to use it to perform K-Fold Validation. The processing time to perform cross validation is very long. Therefore, this dataset was used to save time in doing k-fold validation.

Consequently, the next step is split the data into train dataset and test dataset. The data is transformed to vectors. It would avoid the feature extraction methods to learn features from the

test set. There were 4 types of training and testing dataset. It would ensure the results from different feature extraction methods would not overlap with each other. Bag of Words has two training and testing datasets which were formed from partial data (“x_partial”, “y_partial”) and full data (“x”,“y”). Meanwhile, TF-IDF has one dataset that was formed from full data (“x”,“y”). These datasets mentioned above were for testing purposes and it would not be used as input for the final model. The fourth training set and fourth testing dataset were the original data and it would be used in the final model.

No	Feature Extraction Method	Type of Data used	Names of Dataset
1	Bag of Words	Partial Data	x_train_bow_p, x_test_bow_p, y_train_label_bow_p, y_test_label_bow_p
2		Full Data	x_train_bow, x_test_bow, y_train_label_bow, y_test_label_bow
3	TF-IDF	Full Data	x_train_tfidf, x_test_tfidf, y_train_label_tfidf, y_test_label_tfidf
4	None	Full Data	x_train, x_test, y_train_label, y_test_label

Table 4.1.1 Types of Dataset Used to Train the Model

4.1.3.2 Process of model training and evaluation

After splitting the data, cross validation was only performed on the partial data (“x_partial”, “y_partial”) that undergoes BOW. The type of cross validation method was K-Fold Cross Validation. It was chosen as it works well on a balanced dataset. Different algorithms such as Multinomial Naïve Bayes, Support Vector Machine, decision tree, and random forest algorithm were implemented to find a suitable algorithm.

Multinomial Naïve Bayes was proposed as one of the algorithms because it is easy to utilize. It does not need a lot of training data to find the parameters. This algorithm was appropriate to use in this proposed project because of the limited datasets.

Support Vector Machine was the most common algorithm used for the classification task. This algorithm will work well in a limited dataset. SVM also has kernels that can be utilized when the datasets are not linearly separable. The parameters will be set accordingly to get the best results. A linear kernel is used in this model because it works well on classification of text.

According to Waseem, the decision tree uses the if-then rules, which are equally exhaustive and mutually exclusive in classification [21]. When a new feature is found it will be split into a group, it will go on until there is no new feature that can be split. This is suitable for this proposed project because it has a lot of different types of words.

A random forest algorithm is an ensemble of Decision Trees because it combines multiple decision trees to increase the precision. It also has the best accuracy when it is compared with other algorithms.

The parameters of algorithms were set to be the default to ensure it will produce a fair result. The parameters were listed in the table below. Number of folds used for cross validation is 10. The models would be evaluated by looking at the confusion matrix, validation F1-score, validation accuracy score, validation recall score and validation precision score. The table shows the results after performing k-fold validation. All these scores were a mean score.

Machine Learning Algorithms	Results (Mean)
Multinomial naïve bayes	Validation Accuracy = 0.803788 Validation Precision = 0.784746 Validation Recall = 0.805632 Validation F1 Score = 0.794913
Support vector machine	Validation Accuracy = 0.880766 Validation Precision = 0.907295 Validation Recall = 0.832707 Validation F1 Score = 0.868263
Decision tree	Validation Accuracy = 0.853003 Validation Precision = 0.841363 Validation Recall = 0.849026 Validation F1 Score = 0.845033

Random forest algorithm	Validation Accuracy = 0.868211 Validation Precision = 0.882343 Validation Recall = 0.832087 Validation F1 Score = 0.856352
-------------------------	---

Table 4.1.2 Results of 10 fold validation

The random forest classifier has a high accuracy and high recall too. Thus, another way to evaluate the model was to look at the confusion matrix. In this project, 1 was set to represent cyberbullying and 0 was set to represent non-cyberbullying. Since it was targeting cyberbullying classes (target – 1), so the model must have low false negative so that the machine would not predict the cyberbullying as non-cyberbullying. This was due to the objective was to identify cyberbullying message correctly. Therefore, this project is aiming to find models that have low type II error, high precision in looking for negative messages and high accuracy.

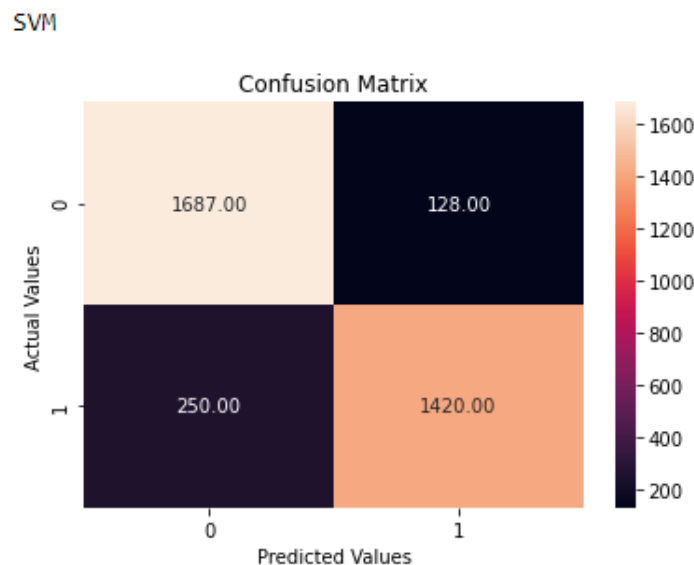


Figure 4.1.3.1 Confusion Matrix of SVM

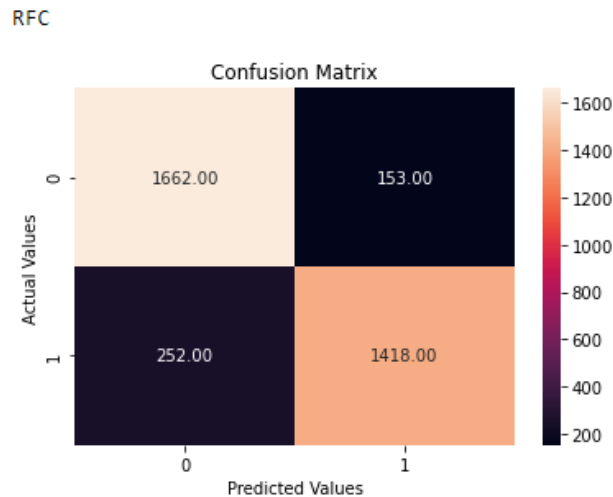


Figure 4.1.3.2 Confusion Matrix of random forest

Based on the confusion matrix of both models, it could be seen that SVM has a lower False Negative (250) when it was compared with random forest (252). Therefore, the machine learning algorithm Support Vector Machine was chosen. Other machine learning algorithms would not be used because the following steps were to compare which vectorization method was better.

The chosen model would be trained again with BOW but utilizing the complete data. To ensure the validation data is not fitted in the model, a new model would be cloned using this code `sklearn.base.clone()` to duplicate a new classifier that has the same parameters and it does not store the validation data. It would act as the base model because it would be compared with the final model. The results were generated after the model was tested with test sets. It would be evaluated with these performance metrics, confusion matrix, precision and recall graph, Cohen Kappa score and Matthews Correlation Coefficient (MCC).

Support Vector Machine
0.9097560975609756

	precision	recall	f1-score	support
0	0.88	0.93	0.91	3245
1	0.94	0.89	0.91	3725
accuracy			0.91	6970
macro avg	0.91	0.91	0.91	6970
weighted avg	0.91	0.91	0.91	6970

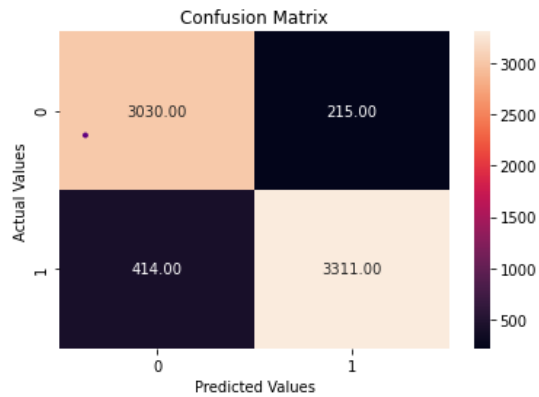


Figure 4.1.3.3 Classification results of Base Model

Based on the picture shown above, accuracy score of base model was high. It was 0.91. The classes also have high recall (0.93, 0.89) and precision (0.88, 0.94). It was important that the recall for class “1” is high. This will ensure the cyberbullying classes would not be misclassified as non-cyberbullying classes. The number of true negatives were low, it was 414.

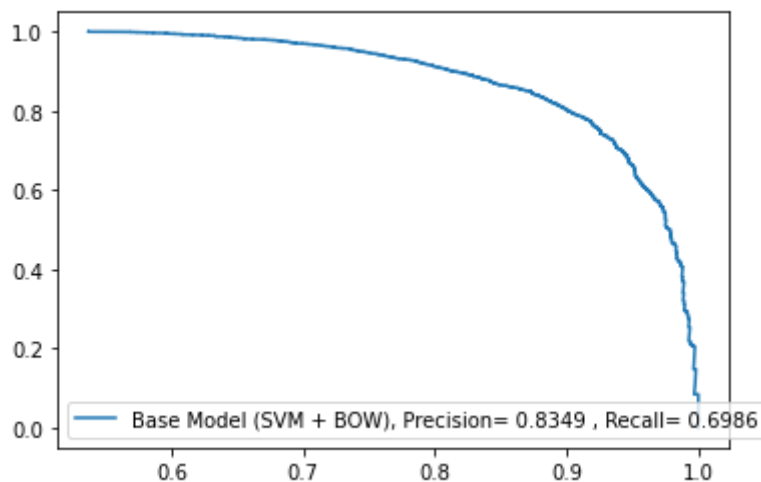


Figure 4.1.3.4 ROC curve of base model

ROC curve stated overall recall score for model was low, the recall score might improve if the model was fined tuned. The Cohen Kappa score for the base model with BOW was 0.8194, while the Matthews Correlation Coefficient (MCC) was 0.8207. If the Cohen Kappa score is

almost 1, it means the classifier is good. It shows the performance of a classifier when it was compared with another classifier that guess randomly based on the frequency of each class. MCC was used to find out the performance of the model in predicting both classes. If the score is near to 1, it indicates that both classes in the model was predicted correctly. Therefore, both scores are good for this base model.

Grid Search was implemented to get the best parameters to create the final model that uses BOW. The grid search uses 5 folds to find the parameters. Two models were created using BOW but the machine learning algorithm used was the same. The first model was a base model that was not fine tuned and the final model was fined tuned. The parameters used for the regularization parameter C was a different combination of values. The correct C score will help the model. The kernel chosen is linear kernel. After performing grid search, the parameters that would create a good classifier are C=0.1 and kernel is linear.

	precision	recall	f1-score	support
0	0.91	0.94	0.92	3245
1	0.94	0.92	0.93	3725
accuracy			0.93	6970
macro avg	0.93	0.93	0.93	6970
weighted avg	0.93	0.93	0.93	6970

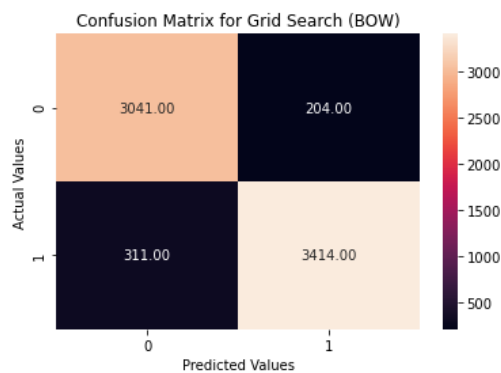


Figure 4.1.3.5 Results after grid search

The accuracy score of the fine-tuned model was even better than the base model. Base model has accuracy score of 91% while the fine-tuned model has an accuracy score of 0.93. The classes in the fine-tuned model also have higher recall score (0.94, 0.92) and precision score (0.91, 0.94). It could be seen that it performed better than the base model because recall score of individual classes from the base model were (0.93, 0.89). It was important that the recall score for class “1” is high. This will ensure the cyberbullying classes would not be misclassified

as non-cyberbullying classes. The number of true negatives of the fine-tuned model decreased and it was only 311 samples. There were a total number of 3414 samples that was labelled as true negatives in the base model.

Another vectorization method was used, it was TFIDF. It will undergo the same process as mentioned above to generate a base model and a final model. However, k-fold validation was not performed.

```
Support Vector Machine
0.9164992826398852
      precision    recall  f1-score   support

     0       0.89      0.94      0.91       3245
     1       0.94      0.90      0.92       3725

 accuracy          0.92       6970
 macro avg         0.92      0.92      0.92       6970
 weighted avg      0.92      0.92      0.92       6970
```

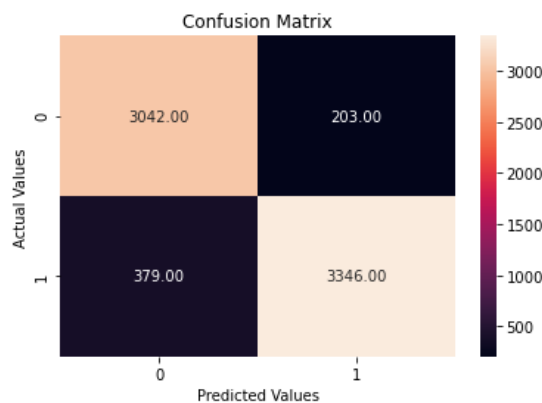


Figure 4.1.3.6 The results of base model with TFIDF

It was observed that the results were similar to the base model with BOW. The number of true negatives was lower, it was 379, it was lower than the base model with BOW. The accuracy score of the base model was high. It was 0.92. The classes also have high recall (0.94, 0.90) and precision (0.89, 0.94).

Grid Search was implemented to get the best parameters to create the final model that uses TFIDF. The grid search uses 5 folds to find the parameters. Two models were created using TFIDF but the machine learning algorithm used was the same. The first model was a base model that was not fine tuned and the final model was fined tuned. The parameters used for the regularization parameter C was a combination of different values. The kernel chosen is linear

kernel. After performing grid search, the parameters that would create a good classifier are $C=1$ and kernel is linear.

	precision	recall	f1-score	support
0	0.91	0.93	0.92	3245
1	0.94	0.92	0.93	3725
accuracy			0.92	6970
macro avg	0.92	0.93	0.92	6970
weighted avg	0.93	0.92	0.92	6970

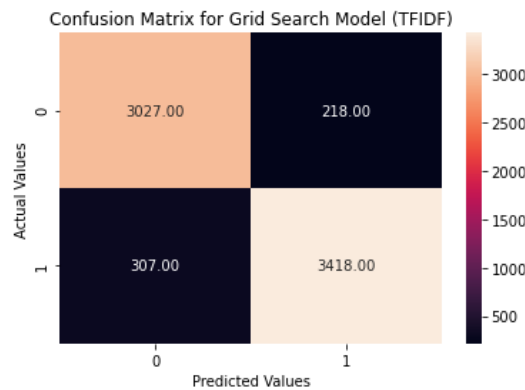


Figure 4.1.3.7 Results after grid search (TFIDF)

Based on the results shown above, the accuracy score of the fine-tuned model with TFIDF was same as than the base model. The fine-tuned model also have higher recall score (0.93, 0.92). It could be seen that it performed better than the base model because recall score of individual classes from the base model were (0.94, 0.90). It was important that the recall score for class “1” is high. This will ensure the cyberbullying classes would not be misclassified as non-cyberbullying classes. The number of true negatives of the fine-tuned model decreased and it from 379 to 307 samples. There were a total number of 3418 samples that was labelled as true negatives in the base model and it was better than the base model with TFIDF.

Finally, a ROC curve was plotted to see if the fined-tuned model with BOW was better than the fined-tuned model with TFIDF. Since the machine learning algorithm used would be the same.

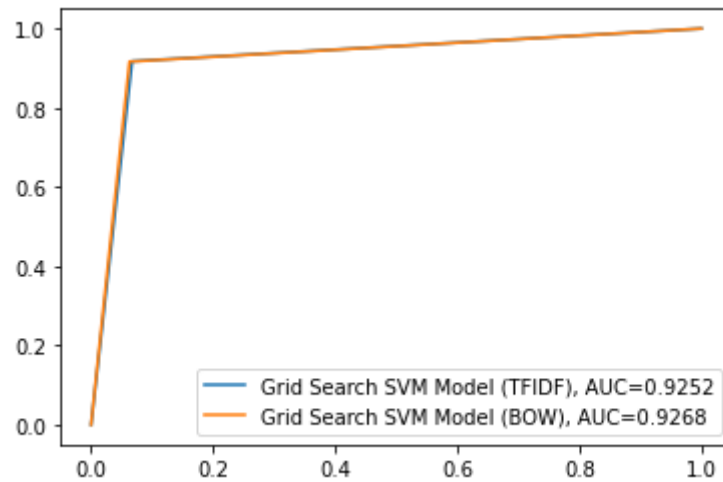


Figure 4.1.3.8 ROC curve of both fined tuned model

Based on the results, the fine-tuned model with BOW has a higher AUC score. This shows that the model with BOW was better at differentiating both classes. Therefore, it was chosen to be the final model in his project.

A precision recall graph for both models with BOW was plotted to see the improvement of the scores. It was observed that the recall score of the model improved and it was slightly higher than the base model.

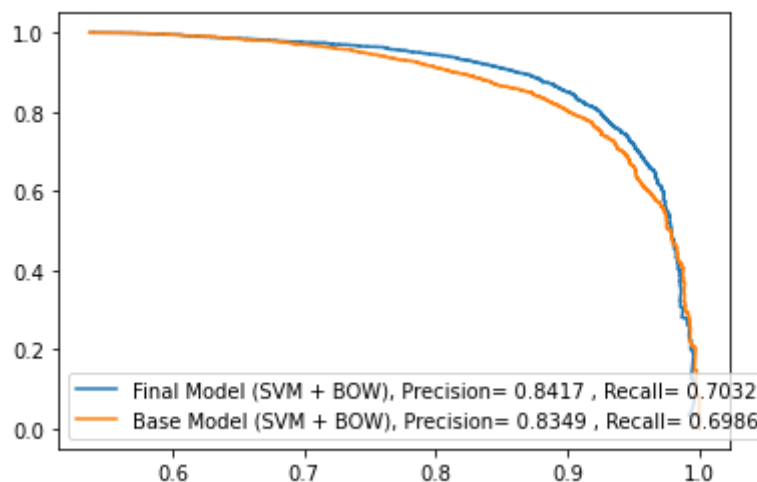


Figure 4.1.3.9 Precision Recall Graph for both models with BOW

4.1.4 The Final Machine Learning Model

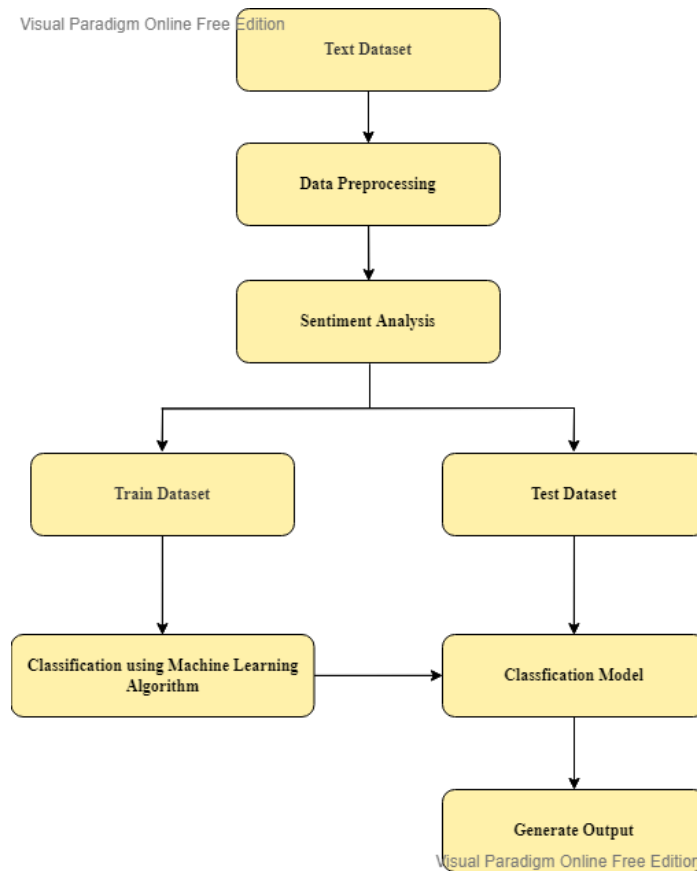


Figure 4.1.4.1 Block Diagram of the Cyberbullying Classifier

All codes to build a machine learning model were written on Jupyter Notebook. It was written in a separate program. The data was pre-processed to remove all unwanted features. The text was converted into lowercase. Tokenization was also applied to pre-process the data. It will separate the text into individual terms. Besides that, stop words were removed from the datasets. Lemmatization with a Part of Speech tag was implemented to normalize the text.

The target feature of the data was created by labelling all the messages with TextBlob. The target features would be “Negative” and “Positive”. However, it might not be efficient during model training. Thus, the labels were converted to numerical form. “1” represents negative samples and “0” represents positive samples. Then, these target features would be used.

Finally, Support Vector Machine and feature vectorization method, Bag of Words model would be chosen to create a model to load into the web application.

The final model will be pickled into a joblib file. It was known as “text_classification.joblib”. The file will be loaded into the web application. The final model would be able generate output. The model would be trained to learn that negative classes were cyberbullying and positive samples were non-cyberbullying class. It will classify the sentence into cyberbullying and non-cyberbullying and display the classification results. It would also be able to predict the probability score of the sentence. The probability score would be utilized in the web application to find the likelihood of the message.

4.2 Web Application

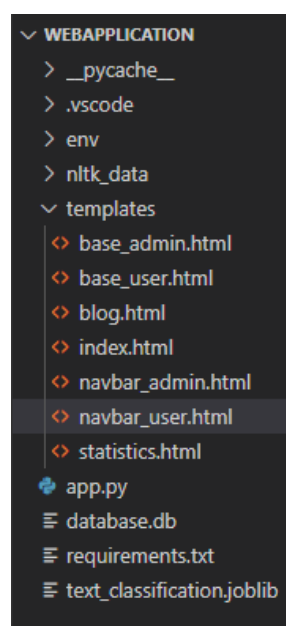


Figure 4.1.4.1 Structure of the Web Application

Web application loaded with the cyberbullying classifier will be able to predict input. The model was pickled because it was a method to save trained models. The model was named as “text_classification.joblib”.

The purpose of creating this web application was to test if the machine learning model was working well with different types of input. It was also used to simulate a scenario where cyberbullying activities that might occur in the social media site. It would have stored all the results in the database (database.db) and the results would be queried for the admin to analyse the result. The results include the classification of text, abusive words found in the sentence and the likelihood of cyberbullying based on the text. The corpus of abusive words was stored

in the file folder named “nltk_data”. A requirements file was created so that the user can install all the packages needed to run the program.

The backend engine was built using Flask [14]. Flask is a web framework and it is used to build a web application using Python [15]. It would process the request sent by the user. The “app.py” file was the main python file that would execute all the functions created and web pages. The web application was built in a virtual environment called “env”. This would store all requirements in the project folder, it would not be installed on the system itself. It makes it easy to transfer the file to another device. This web application has 3 main web pages, “Index.html”, “Blog.html” and “Stats.html”. Each of the web page has their own purpose. It would be explained in the sections below.

The flask application has a secret key to protect the information that was submitted in the form. WT Forms (WTF) module in Flask was used to design the forms. It was easier to design form as a class of the Form would be defined, and it could be reused in other web pages. The web application was written in HTML, python. The layout of the web pages was designed with bootstrap. It uses Jinja web template to create a base template. This template was useful in this project because it would ensure the layout of the web page would look the same. These base templated would be embedded into the three main pages. It uses a `{{ % % }}` to apply the settings of the base template into the targeted web pages. Two base templates were created for the blog and index page, it was known as “base_user.html” and “navbar_user.html”. The “base_user.html” page would contain the default layout of the pages. The “navbar_user.html” would have the code to display the navigation bar to go the two main web pages which were “Blog.html” and “Index.html”. Thus, it would be embedded into the “base_user.html” page. Another two templates were created for the admin, it was known as “base_admin.html” and “navbar_admin.html”. This was because the “Stats.html” page was not to be accessed by the website users. Thus, “navbar_admin.html” that has a navigation bar to go all three pages would be created in the “Stats.html” page.

4.2.1 Creating a Database (SQLite)

In this project, SQLAlchemy extension in Flask was used to make a simple database to save all the input data from user. It is a Python ORM (Object Relational Mapping) library. The database created was SQLite. With the use of SQLAlchemy ORM, the attributes would be saved.

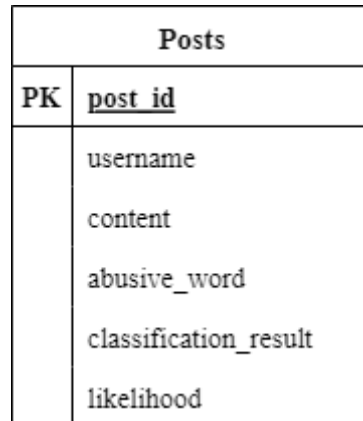


Figure 4.2.1.1 ERD of Database

Table created in the database was called Posts. It would store the ID of the post, content of the post, username of the website user, label of the website user, abusive word in the post, likelihood of cyberbullying based on the post. It would store the data collected from the “Blog” page. The description of the attributes was listed in the table below.

Attribute	Data Type	Description
post_id (Primary Key)	Integer	It is the primary key. It is the id of each post.
username	String (50)	It will store the usernames that was collected from form. It is the username of the poster.
content	Text	It will store the message that was collected from form. It is the message that user wants to post on the blog
abusive_word	String (50)	These are the words that have negative meaning and it might hurt someone’s feeling when it was written. It will store all abusive words found in the sentence.

classification_result	String (50)	The results generated from the cyberbullying classifier. It will store all classification results.
likelihood	String (50)	It represents the rating of the cyberbullying message. If it was high, it means the message is likely to be a cyberbullying message. It will store the likelihood of cyberbullying of the message.

Table 4.2.1 Description of Attributes

4.2.2 Wireframe of Web Pages

4.2.2.1 Index.html (Home page)

Home	Blog
-------------	-------------

Cyberbullying Detection

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Type your message here :

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Results

Abusive Words :	Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Classification Results :	Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Likelihood :	Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Figure 4.2.2.1 Wireframe of Index.html

This web application has 3 web pages. The first web page was called index.html also known as “Home”. A navigation bar was created on top to access all pages that the user was allowed to use. This page would be accessed by the website user and the admin. The purpose of building

this page was to test the cyberbullying classifier. It will output the classification result. It would also have the following functions to detect abusive words contained in the text and the likelihood of cyberbullying. These two functions were created to support the claim that the message might contain cyberbullying content. It would display the abusive word and likelihood of cyberbullying to the user. The message submitted to the form would be passed into the classifier for classification. The messages would be passed into the functions to find out the abusive word and likelihood of cyberbullying. The form was designed with WT Forms (WTF) module in Flask. It validates the input and does not allow the user to submit empty data. The data that were tested in this page would not be stored into the database. . It would allow the user to resubmit the form. The data of the form would be cleared after the user submits the form. It would not allow the user to submit an empty form.

4.2.2.2 Blog.html

Home	Blog	
-------------	-------------	--

Blog

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Create Post
Username

Content

All Posts

Username
Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Username
Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Username
Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Figure 4.2.2.2 Wireframe of Blog.html

This page would be accessed by the website user and the admin. Users were allowed to create and view the blog posts on web page known as “Blog”. A navigation bar was created on top to access all pages that the user was allowed to use.

In this web page, all the information collected from the form would be saved in the database. Functions would be created to query or save the data from the database and get input from user. For example `blogPost()`, this function would have a form that would accept user input and saves

the input into the database. The form created from the functions would be sent to the url that was mapped. For example, the `blogPost()`, would display the form that accepts input from user. The Hypertext Transfer Protocol (HTTP) of the form uses post method because it will send data to the backend engine [14]. Once the request was received, the backend engine would save the results in the database. The functions such as find abusive word and determine the likelihood of cyberbullying of message would be used to fill up the data in the database. It would also classify the message and save the result in the database. When the user sends a POST request, the pre-trained model was used to predict the input, ensuring that the model has not been trained again [14]. These results would be used in the “Stats” page. It would allow the user to resubmit the form. The data of the form would be cleared after the user submits the form. It would not allow the user to submit an empty form.

All the blog posts that were posted by the users would be displayed below. It will display the username and the content of the posts. The other results would be hidden from user’s view and only admin can read the results.

4.2.2.3 Stats.html

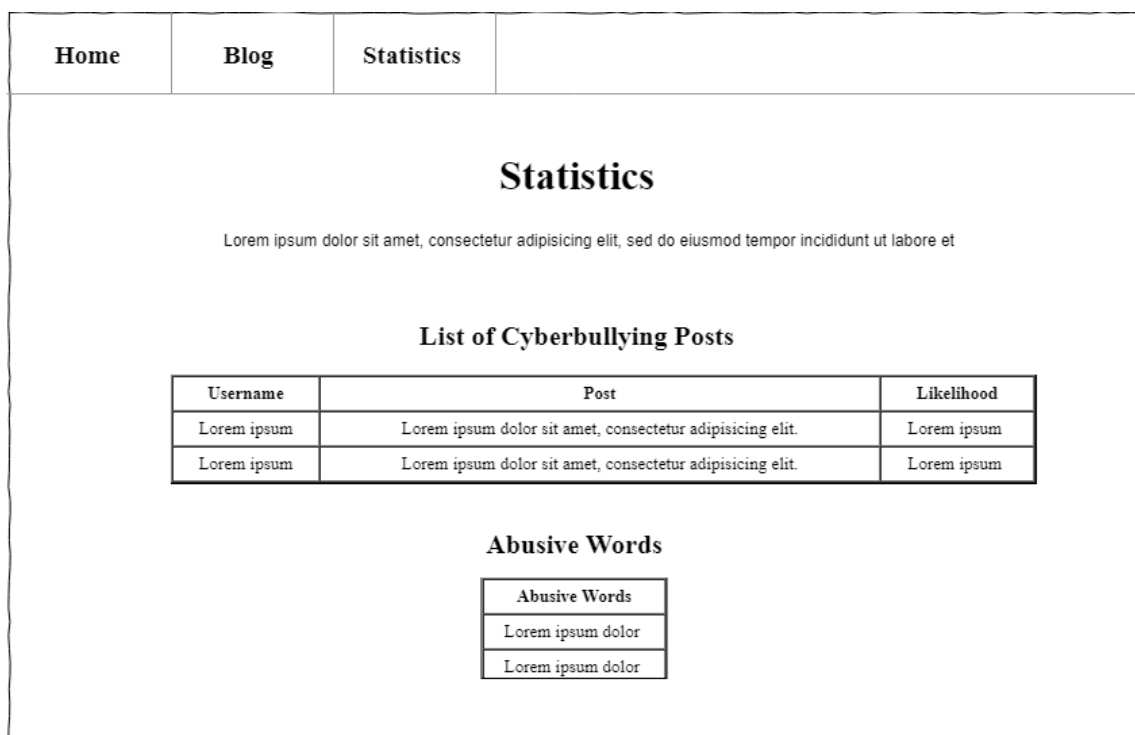


Figure 4.2.2.3 Wireframe of Stats.html

A page known as “Statistics” was specifically designed for the admin to track and monitor the cyberbullying activities. It would not be visible to the user. A navigation bar was created on top to access all pages. The admin would be able to access detailed information such as the post that was classified as cyberbullying and the name of the user that posted the post. Therefore, the admin need to type “/stats” behind the web page link to view the results.

The first table would show the lists of posts that are classified as cyberbullying. All these data were queried from the database and displayed in the table using a for loop. The posts that were classified as non-cyberbullying would not be displayed on this page. The table would also consist of the name of the user and the likelihood of the cyberbullying for the posts.

The second table shows the list of abusive words found in each post that was classified as cyberbullying. It will show all the distinct abusive words, duplicated words would not be displayed in this table.

4.2.3 Building a Corpus to Store All Abusive Words

In this project, a function was added into the web application to determine the abusive word in a sentence. The corpus was built on Jupyter Notebook. The data that used to build the corpus was `cyberbullying_tweets.csv`. The total number of rows was 47692. The same data was also utilized to train the model. The corpus would store all the abusive words that was found in the dataset. The purpose of building this corpus is to collect all negative words that may hurt a person’s feelings. The data was preprocessed to eliminate the stop words, symbols, Uniform Resource Locator (URL), usernames, punctuations and non UTF-8 or ASCII characters. The data undergoes further preprocessing steps such as tokenization and lemmatization.

Two methods were used to find abusive words in the dataset. The methods are using TextBlob and VADER. Both methods were used to determine the polarity of each word. The results generated from both methods were compared to obtain a valid list of abusive words. The words were categorized into three categories which are Positive, Negative and Neutral. A python list is created for each category, such as “positiveWords”, “abusiveWords” and “neutralWords”. The range of values to determine the polarity score of word in TextBlob and compound score of word in VADER is equivalent with the value used in the data preprocessing phase that was implemented in final year project 1. The polarity score and compound score will determine the

category of the word. For loop and while loop was implemented to assign each word into their respective list. All duplicated values from the list would be removed and a new text file was created to store these values. The processing time to categorize the complete data with VADER was too long. Therefore, the data was split into different range of values. Hence, different sets of data were created using VADER. The table below are the text files generated.

Method	Filename of Text File	Range of data
TextBlob	abusiveWordsTEXTBLOB.txt positiveWordsTEXTBLOB.txt neutralWordsTEXTBLOB.txt	All values in the csv file
VADER (First set)	abusiveWordsVADER1.txt positiveWordsVADER1.txt neutralWordsVADER1.txt	1 to 10000
VADER (Second set)	abusiveWordsVADER2.txt	10001 to 20000
VADER (Third set)	abusiveWordsVADER3.txt	20001 to 30000
VADER (Fourth set)	abusiveWordsVADER4.txt	30001 to 40000
VADER (Fifth set)	abusiveWordsVADER5.txt	40001 to 47692

Table 4.2.2 The Name of the Files Generated

The first set of text file from VADER was analyzed to see if any words are misclassified into the wrong categories. Since there were a lot of sets of data, only the first set was analyzed for all three categories. Then, the finalized set was the combination of all “abusiveWords” text files generated previously. The duplicated values in the file would be removed. It was known as “AbusiveWords (final).txt”. It can be found under the nltk_data folder.

The benefit of building a corpus was the content in the corpus can be edited using Python built-in functions. Functions such as capitalize() and upper() were applied to transform the words found and it would increase the variety of words. This function would be used in 2 web pages which were home and statistics. In the index page, it will output the abusive words to the user but these words would not be stored in the database. The data collected from the blog page will be analyzed in the statistics page to create a table to display the abusive words used by the user.

4.2.4 A Function to Determine the Likelihood of Cyberbullying

It uses the results from the classifier to find out the likelihood of cyberbullying. The `predict()` and `predict_proba()` function were used to get the classification result of the message and probability of the message being a cyberbullying message. Referring to the classification results generated, when the probability calculated was high, it means it is a cyberbullying case. When the probability was low, it means it is a non-cyberbullying case. The predictions represent the classification results. When the prediction was “1”, it means the message was classified as cyberbullying. When the prediction was “0”, it meant the message was classified as non-cyberbullying. The range of values for likelihood were very high, high, low, and very low. If the probability is 90 - 100 percent and prediction is “1”, then the likelihood of cyberbullying is Extreme High. If the probability is 70 - 89 percent and prediction is “1”, then the likelihood of cyberbullying is Very High. If the probability is 50 - 69 percent and prediction is “1”, then the likelihood of cyberbullying is High. If the probability is 30 - 49 percent and prediction is “0”, then the likelihood of cyberbullying is Low. If the probability is 10 - 29 percent and prediction is “0”, then the likelihood of cyberbullying is Very Low. If the probability is 0 - 9 percent and prediction is “0”, then the likelihood of cyberbullying is Extremely Low. It will prompt a message such as “Likelihood of cyberbullying is very high” when all the conditions are met. It will also prompt error message if the conditions were not met.

Polarity	Probability (%)	Likelihood of cyberbullying
Negative	90 - 100	Extremely High
	70 - 89	Very High
	50 - 69	High
Positive	30 - 49	Low
	10 - 29	Very Low
	0 - 9	Extremely Low

Figure 4.2.4.1 Likelihood of cyberbullying cases

4.2.5 Final Design of all Web Pages

Home Blog

Blog

Users can create posts to simulate a social media site. The posts that are created would be shown below.

Create Post

Username

Content

Create Post

All Posts

darren
I hate you

su yen
You are so friendly, i like you.

Figure 4.2.5.1 Design of the Blog Page

Home Blog

Cyberbullying Detection

Users can input a message to find out if it is cyberbullying text. It will also display the abusive words found and the likelihood of the cyberbullying of the message.

Type your message here :

Classify Text

Results

Abusive Word:

Classification Results:

Likelihood:

Figure 4.2.5.2 Design of the Home page

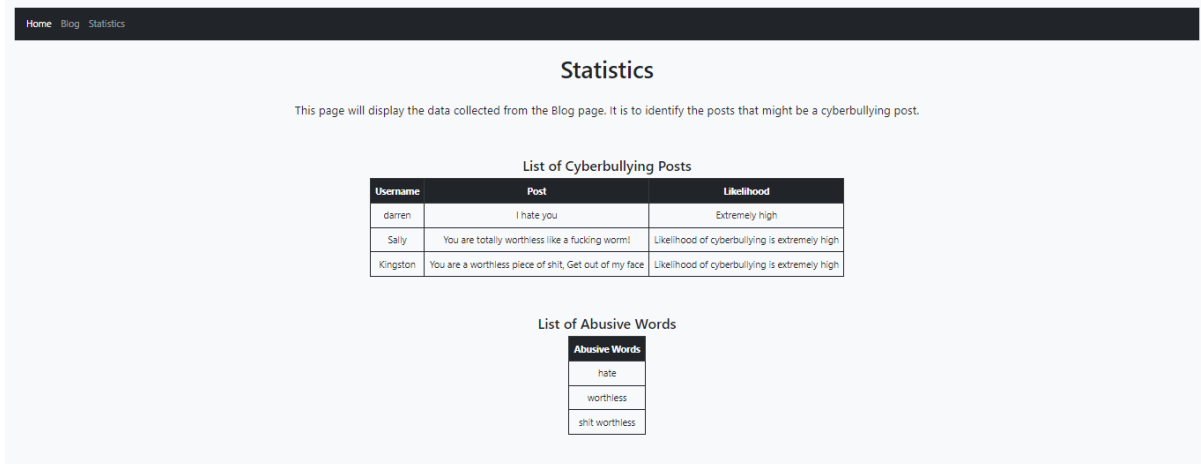


Figure 4.2.5.3 Design of the Statistics Page

The color scheme was different in the statistics page because it can only be accessed by the admin.

Chapter 5 System Implementation

5.1 Hardware Setup

Description	Specifications
Model	Asus X407UF
Processor	Intel Core i5-8250U
Operating System	Windows 10
Graphic	NVIDIA GeForce MX130 2GB GDDR5
Memory	8GB RAM
Storage	1TB

Table 5.1.1 Hardware used in the project

5.2 Software Setup

Development	Software Tools
Programming Language	HTML, Bootstrap, JavaScript, Python
Software	Microsoft Visual Studio Code, Jupyter Notebook, Anaconda
Web Framework	Flask
Database	SQLite from SQLAlchemy

Table 5.2.2 Software used in the project

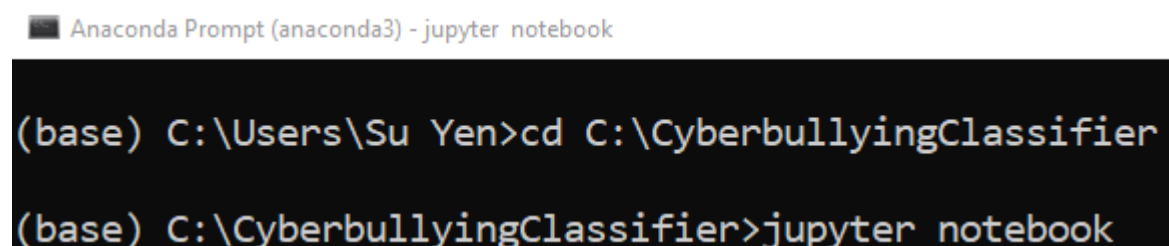
HyperText Markup Language (HTML) is the programming language used to write the web application. It is applicable in a web browsers like Google, Safari and more. Bootstrap were

used to design the style of the web application. The language python is used for web development and machine learning.

Jupyter Notebook is a web application that was used to write codes in Python language. Anaconda is a program that allows the user to execute python files in ipynb format. Therefore, all programs written in Jupyter Notebook would be opened using Anaconda prompt. Flask is from a third-party Python library that is used to create web applications. It is to integrate the classifier with the web application. SQLAlchemy extension in Flask was used to make a simple database to save all the input data from user. It is a Python ORM (Object Relational Mapping) library. The database created is SQLite. With the use of SQLAlchemy ORM, the attributes would be saved in the database. WT Forms (WTF) module in Flask is used to design the forms. Jinja was also used to design the web application. It has a lot of templates that could be used for web development. Microsoft Visual Studio Code is an integrated development environment and code editor to write and debug web applications.

5.3 System Settings and Configuration

The programs to develop machine learning model was written on the jupyter notebook. All the files would be store in project folder called Cyberbullying Classifier. The requirements to run the code were listed in the requirement.txt file. It needs to be run to make sure all the packages were installed in the anaconda prompt. In this project, anaconda prompt was required to start the program. It is required to copy the file path into the anaconda command prompt. As shown below, the project directory needs to be specified to launch the notebook such as “cd C:\CyberbullyingClassifier”. To start the program, “jupyter notebook” must be typed.



```
Anaconda Prompt (anaconda3) - jupyter notebook
(base) C:\Users\Su Yen>cd C:\CyberbullyingClassifier
(base) C:\CyberbullyingClassifier>jupyter notebook
```

Figure 4.2.5.1 Anaconda prompt

The user will go to the web page on the browser.

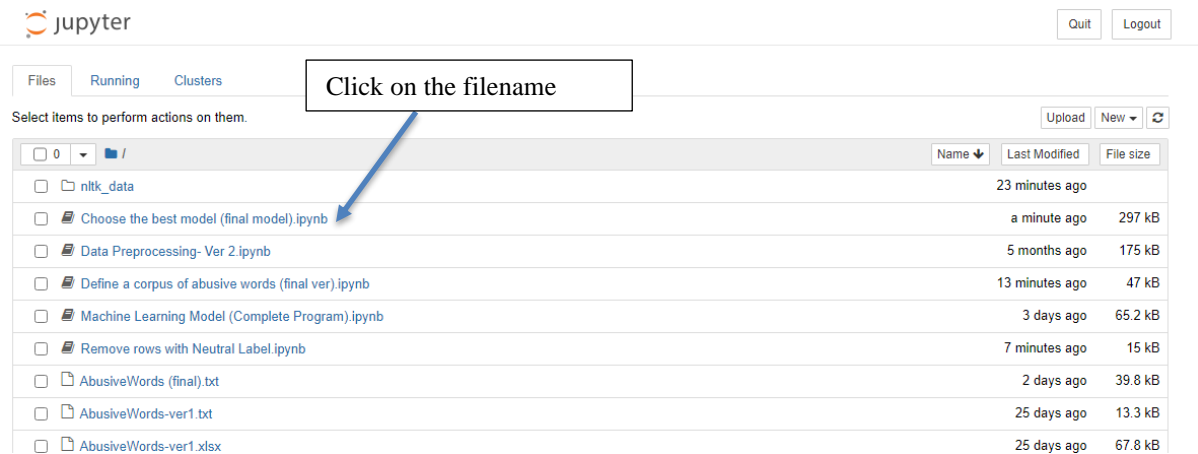


Figure 4.2.5.2 Jupyter notebook home page

It have all the files of the project folder. To start the program, need to click on the name of the file. Then, it will be redirected to the codes.

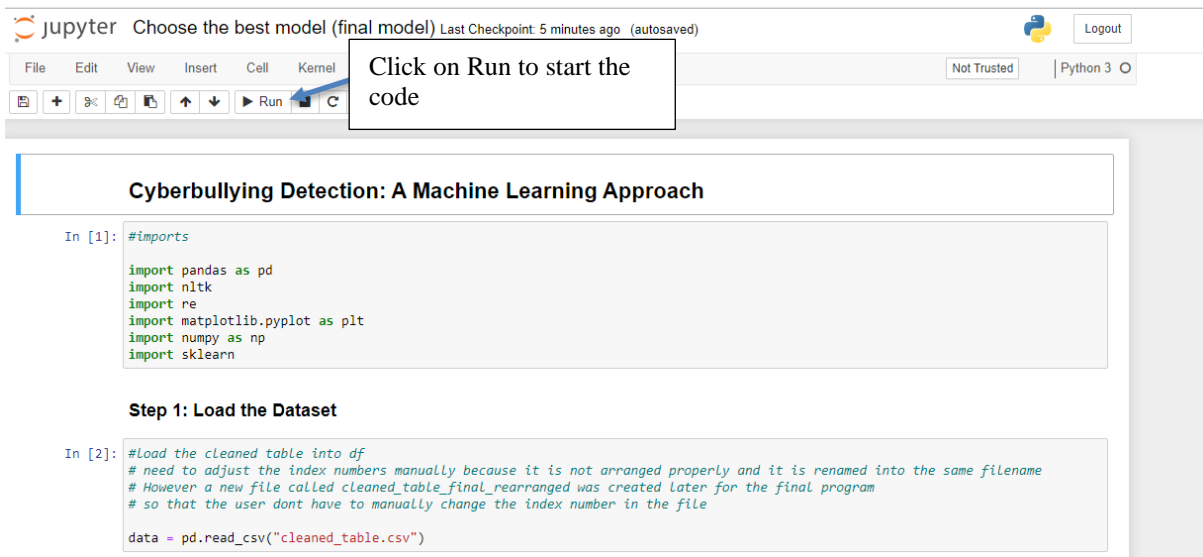


Figure 4.2.5.3 The page that has the codes

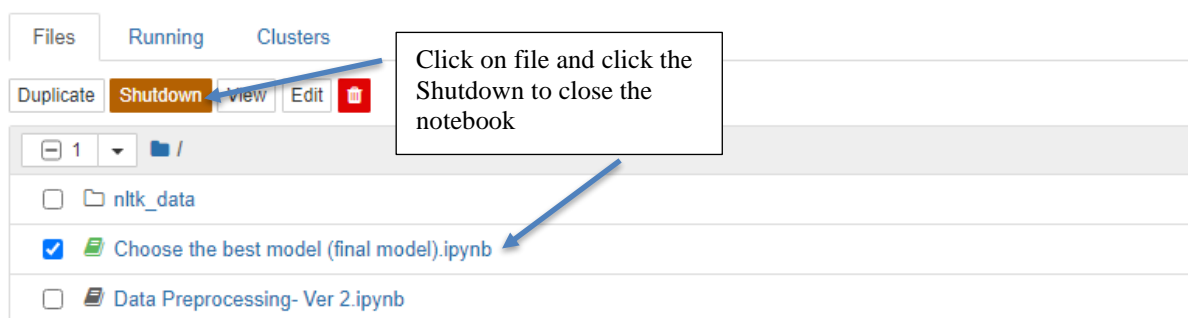


Figure 4.2.5.4 Shutdown the notebook

Cyberbullying Detection

Users can input a message to find out if it is cyberbullying text. It will also display the abusive words found and the likelihood of the cyberbullying of the message.

Type your message here :

I think you are stupid

Classify Text

3. Type the message

2. Classify the text

Results

Abusive Word: stupid

Classification Results: Cyberbullying

Likelihood: Likelihood of cyberbullying is extremely high

1. Results generated

Figure 4.2.5.1 Message typed into the form

Message must be typed into the form to classify the text. The results generated would be shown below. The form must be filled before submitting it, it would show error message if it was not filled. It applies to all the forms in the web application.

Cyberbullying Detection

Users can input a message to find out if it is cyberbullying text. It will also display the abusive words found and the likelihood of the cyberbullying of the message.

Type your message here :

Classify Text

Please fill out this field.

Results

Abusive Word: stupid

Classification Results: Cyberbullying

Likelihood: Likelihood of cyberbullying is extremely high

Figure 4.2.5.2 Error message shown

When accessing the second web page, user can click on the “Blog” in the navigation bar or copy paste this link [/blog_post](#) at the end of the link. This link would be generated when the app was running on the terminal in Visual Studio Code.

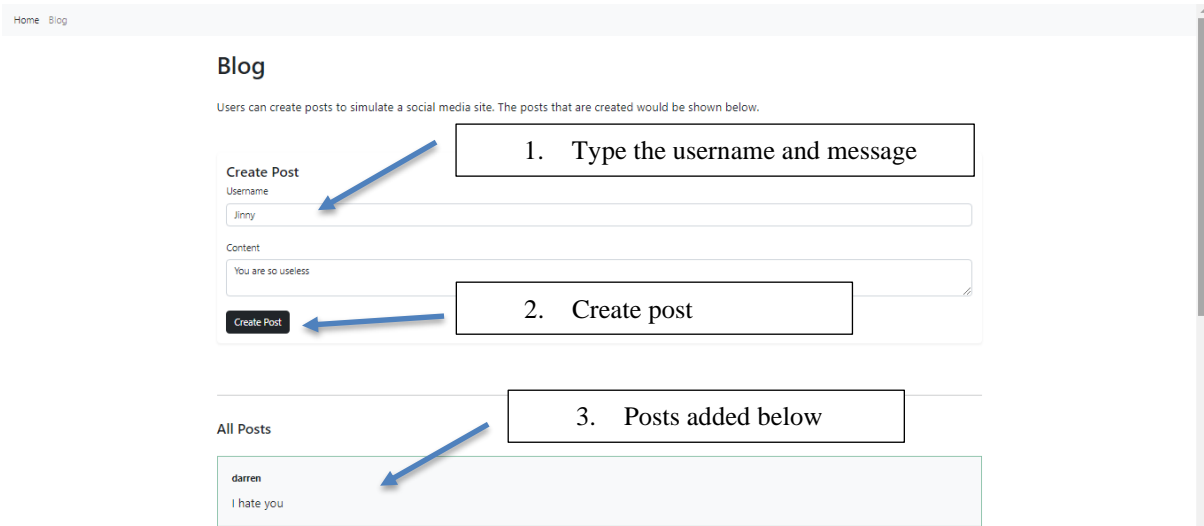


Figure 4.2.5.3 Create Blog Post

User need to input their username and content to create a blog post.

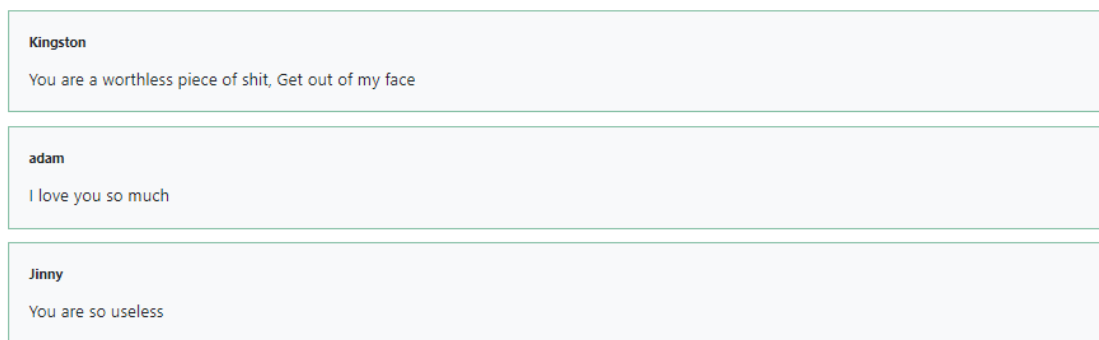


Figure 4.2.5.4 Show Blog Post created

The blog posts created would be shown below. To access the third page, the admin will have to write this link <http://127.0.0.1:5000/stats>.

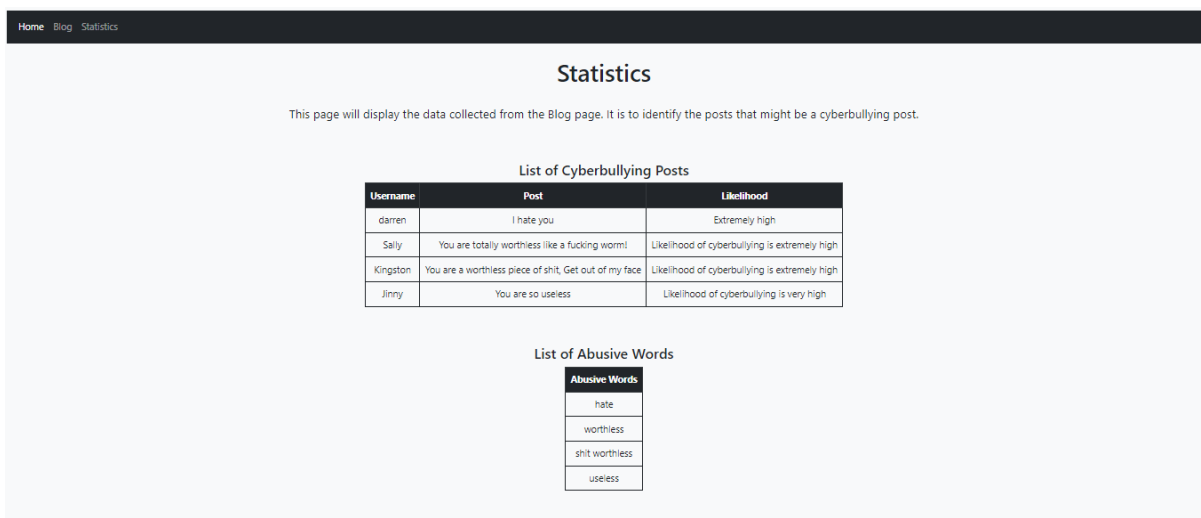


Figure 4.2.5.5 Statistic page will show the updated data

The page will show all the data that was collected from the form. It only shows the cyberbullying posts and abusive word. The details of the table would be updated automatically when the user submits the form. It has a navigation bar that allows the admin to access all three pages.

Chapter 6 System Evaluation And Discussion

6.1 System Testing and Performance Metrics

Black box testing was used to test the web application. This testing technique does not involves inspecting the code but looks at the output generated.

6.2 Testing Setup and Result

Test Case	Test Case Description	Flows	Expected Result	Actual Outcome	Pass/Fail
Input Text	User can input the text.	<ol style="list-style-type: none"> 1. User input message in the form. 2. User submits form. 	Form was submitted. Empty form must not be submitted and prompt error message.	Form was submitted. Prompt error message when empty was submitted.	Pass
Classify Text	User can classify the text after submitting the form.	<ol style="list-style-type: none"> 1. User click on the “Classify Text” button. 2. System classifies the text. 3. System calls the function to find abusive words in the sentence. 4. System calls the function to determine the likelihood of the cyberbullying to the admin or website user. 	All the results such as classification result, abusive word, and likelihood of cyberbullying were generated.	All the results such as classification result, abusive word, and likelihood of cyberbullying were generated.	Pass

		6. System displays the results to the admin or website user.			
View Classification Result	User can view the results after submitting the form.	1. System displays abusive words found in the sentence, classification of the sentence and the likelihood of the cyberbullying to user. 2. User views the results.	Abusive words found in the sentence, classification of the sentence and the likelihood of the cyberbullying was displayed to the user.	Abusive words found in the sentence, classification of the sentence and the likelihood of the cyberbullying was displayed to the user.	Pass
Create Blog Post	User can create a blog post.	1. User input username and message. 2. User creates post. 3. System saves the username and message in the database. 4. System classifies the sentence. 5. System calls the function to find abusive words in the sentence. 6. System calls the function to determine the likelihood of the cyberbullying to the admin	User can create the blog post and submit it. All the data would be saved in the database.	User can create the blog post and submit it. All the data would be saved in the database.	Pass

		<p>or website user.</p> <p>7. System saves the abusive words in the database.</p> <p>8. System saves likelihood of the cyberbullying to the admin or website user.</p> <p>9. System will clear the form and allow admin or website user to resubmit the form.</p>			
View Blog Post	User can view all the blog posts created.	<p>1. User views blog post after submitting the form.</p> <p>2. System displays the post created to the user.</p> <p>3. System shows all the posts that was created.</p>	User can see all the blog posts. The username and content was shown.	User can see all the blog posts. The username and content was shown.	Pass
View Blog Activities	User can view the data from the database.	<p>1. System retrieves all the data from the database.</p> <p>2. System displays a table that has posts that are classified as cyberbullying, the username that posted the post and likelihood of</p>	User can view all the posts that are classified as cyberbullying and a list of abusive words.	User can view all the posts that are classified as cyberbullying and a list of abusive words.	Pass

		cyberbullying based on the post.			
		4. System displays table that consist of abusive words.			

Table 6.2.1 Results of all Test Cases

6.3 Project Challenges

These are the implementation issues that I have encountered while doing this project. There are not a lot of public datasets available for cyberbullying on social media. Most of the datasets are related to customer reviews, movie reviews and more. Therefore, it is hard to find a suitable dataset. The dataset in this project has a lot of tweets that are related to cyberbullying and it has a diverse topic. However, there are many text data that is not useful for the model such as hyperlinks and usernames. So, the data must be cleaned properly before using it to train the model. Some of the codes used for data preprocessing might cause some errors. Thus, it takes time to solve it and clean the data. The code cannot extract the string from the text such as usernames from the dataset. There are errors in my codes such as cannot break up the strings into tokens when all data is used. The above errors were solved after doing research online. During the execution of VADER to generate a list of abusive words, the processing time was very long if the whole dataset was used. Thus, the dataset was split into different sets to make sure the RAM of the computer are not extensively used. The web application was not deployed because it costs money to deploy the web application on the Microsoft Azure.

6.4 Objectives Evaluation

All the objectives in this project were met. The main goal is to correctly predict the message that belongs to the cyberbullying class. The results shows the precision score of identifying cyberbullying text was high, it was 0.83, and is close to one. The objective was to test if the cyberbullying classification model would work well in the web application to detect the sentence with words that has abusive, offensive, or harmful meaning. The results shows that it can work well in the web application and all data were saved in the database. This project aims to find the most suitable machine learning algorithm that can correctly predict a message that has a negative meaning. Support Vector Machine with an accuracy of 0.93 was built to classify the text. It also has high precision score and good recall score. It could work well on test sets.

Chapter 7 Conclusion and Recommendation

7.1 Conclusion

Cyberbullying cases are increasing as more people are using the Internet. Targeting someone online is easier because they can hide their identity by not revealing their names or profile pictures. The majority of victims are aware that they are being bullied online, but choose to ignore it. According to the article in Star, it stated that Malaysia is ranked as second place among the countries in Asia [31]. Therefore, it can be seen that the number cyberbullying cases among youth are quite concerning [31]. The goal is to create a classifier that can detect identify cyberbullying text. To classify the text, the proposed solution combines a rule-based approach of sentiment analysis, TextBlob with a machine-learning algorithm, Support Vector Machine to create a model. The results shows that svm has the highest accuracy and the model was fined tuned to get better results. The web application with a database was developed to test the effectiveness of the model if it were to be used on a social media site. The web application would serve as a platform to show that it can detect cyberbullying activities and display the analysis of the results to the user.

7.2 Recommendation

It was observed that some of the abusive words does not exist in the abusive word corpus. Therefore, it can be improved by looking for more sources on the internet to expand the variety of words. Another recommendation was to to adjust the parameters in the BOW model and see if it increases the accuracy of the model. The default parameters were used because the objective of the project was to determine the best machine learning algorithm. The analysis of result in the web application could be improved by listing out the cyberbullies and calculate the number of posts that they posted. The list of abusive word could be displayed using WordCloud. It is a picture that consists of all the words used and it was selected based on the frequency count of the word.

REFERENCES

- [1] UNICEF, “Cyberbullying: What is it and how to stop it,” *www.unicef.org*, Feb. 2022. <https://www.unicef.org/end-violence/how-to-stop-cyberbullying> (accessed Feb. 15, 2022).
- [2] S. Redhu, “Sentiment Analysis Using Text Mining: A Review,” *International Journal on Data Science and Technology*, vol. 4, no. 2, pp. 49–53, Jun. 2018, doi: 10.11648/j.ijdst.20180402.12.
- [3] N. B K, P. Shreya, S. Reddy P, and M. Mohamadi Ghousiya Kousar, “Cyberbullying Detection Using Machine Learning,” *International Research Journal of Engineering and Technology (IRJET)*, vol. 8, no. 8, pp. 507–511, Aug. 2021, Accessed: Mar. 22, 2022. [Online]. Available: <https://www.irjet.net/archives/V8/i8/IRJET-V8I868.pdf>
- [4] H. Rosa *et al.*, “Automatic cyberbullying detection: A systematic review,” *Computers in Human Behavior*, vol. 93, pp. 333–345, Apr. 2019, doi: 10.1016/j.chb.2018.12.021.
- [5] M. Sintaha and M. Mostakim, “An Empirical Study and Analysis of the Machine Learning Algorithms Used in Detecting Cyberbullying in Social Media,” Dhaka, Bangladesh, Feb. 2019. doi: 10.1109/iccitechn.2018.8631958.
- [6] C. V. D, “Hybrid approach: naive bayes and sentiment VADER for analyzing sentiment of mobile unboxing video comments,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 5, pp. 4452–4459, Oct. 2019, doi: 10.11591/ijece.v9i5.pp4452-4459.
- [7] GeeksforGeeks, “Python - Lemmatization Approaches with Examples,” *GeeksforGeeks*, Sep. 04, 2020. <https://www.geeksforgeeks.org/python-lemmatization-approaches-with-examples/>
- [8] J. Wang, K. Fu, and C.-T. Lu, “SOSNet: A Graph Convolutional Network Approach to Fine-Grained Cyberbullying Detection,” Atlanta, GA, USA, Dec. 2020. doi: 10.1109/bigdata50022.2020.9378065.
- [9] N. Roy, “Building a Text Normalizer using NLTK ft. POS tagger,” *Medium*, May 01, 2020. <https://towardsdatascience.com/building-a-text-normalizer-using-nltk-ft-pos-tagger-e713e611db8> (accessed Apr. 01, 2022).
- [10] B. Keen, “Basic Language Processing with NLTK – Ben Alex Keen,” *Ben Alex Keen*, May 06, 2017. <https://benalexkeen.com/basic-language-processing-with-nltk/> (accessed Apr. 01, 2022).
- [11] Parul Pandey, “Simplifying Sentiment Analysis using VADER in Python (on Social Media Text),” *Medium*, Sep. 23, 2018. <https://medium.com/analytics-vidhya/simplifying-social-media-sentiment-analysis-using-vader-in-python-f9e6ec6fc52f> (accessed Apr. 01, 2022).
- [12] M. Kumar Barai, “Sentiment Analysis with Textblob and Vader in Python,” *Analytics Vidhya*, Oct. 20, 2021. <https://www.analyticsvidhya.com/blog/2021/10/sentiment-analysis-with-textblob-and-vader/> (accessed Apr. 02, 2022).

- [13] K. Chen, “Introduction to Natural Language Processing — TF-IDF,” *Medium*, May 24, 2021. <https://kinder-chen.medium.com/introduction-to-natural-language-processing-tf-idf-1507e907c19> (accessed Apr. 02, 2022).
- [14] G. C. Ongko, “Building a Machine Learning Web Application Using Flask,” *Medium*, Feb. 18, 2022. <https://towardsdatascience.com/building-a-machine-learning-web-application-using-flask-29fa9ea11dac> (accessed Apr. 04, 2022).
- [15] A. Dyouri, “How To Create Your First Web Application Using Flask and Python 3 | DigitalOcean,” *www.digitalocean.com*, Aug. 19, 2021. <https://www.digitalocean.com/community/tutorials/how-to-create-your-first-web-application-using-flask-and-python-3> (accessed Apr. 03, 2022).
- [16] MonkeyLearn, “Sentiment Analysis: Nearly Everything You Need to Know | MonkeyLearn,” *MonkeyLearn*, Jun. 20, 2018. <https://monkeylearn.com/sentiment-analysis/>
- [17] IBM Cloud Education, “What is Text Mining?,” *www.ibm.com*, Nov. 16, 2020. <https://www.ibm.com/cloud/learn/text-mining>
- [18] IBM Cloud Education, “What is Natural Language Processing?,” *www.ibm.com*, Jul. 02, 2020. <https://www.ibm.com/cloud/learn/natural-language-processing>
- [19] Readable, “Profanity Word Detector - Text Analysis Tools - Unique readability tools to improve your writing! App.readable.com,” *app.readable.com*. <https://app.readable.com/text/profanity/>
- [20] R. Dwivedi, “How Does Support Vector Machine Algorithm Works In Machine Learning? | Analytics Steps,” *www.analyticssteps.com*, May 04, 2020. <https://www.analyticssteps.com/blogs/how-does-support-vector-machine-algorithm-works-machine-learning>
- [21] M. Waseem, “Classification In Machine Learning | Classification Algorithms,” *Edureka*, Dec. 04, 2019. <https://www.edureka.co/blog/classification-in-machine-learning/#tree>
- [22] Jupyter, “Project Jupyter,” *Jupyter.org*, 2019. <https://jupyter.org/>
- [23] A. O. Christiana, O. S. Oladeji, and A. T. Oladele, “BINARY TEXT CLASSIFICATION USING AN ENSEMBLE OF NAÏVE BAYES AND SUPPORT VECTOR MACHINES,” *GESJ: Computer Science and Telecommunications 2017*, Sep. 2017.
- [24] J. Porter, “‘Face with tears of joy’ is once again the most-used emoji,” *The Verge*, Dec. 03, 2021. <https://www.theverge.com/2021/12/3/22816001/most-popular-emoji-2021-face-with-tears-of-joy> (accessed Apr. 13, 2022).
- [25] D. Munandar, A. F. Rozie, and A. Arisal, “A multi domains short message sentiment classification using hybrid neural network architecture,” *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 4, pp. 2181–2191, Aug. 2021, doi: 10.11591/eei.v10i4.2790.

[26] S.-F. Tu, C.-S. Hsu, and Y.-T. Lu, “Improving RE-SWOT Analysis with Sentiment Classification: A Case Study of Travel Agencies,” *Future Internet*, vol. 13, no. 9, p. 226, Aug. 2021, doi: 10.3390/fi13090226.

[27] L. Li, “Response to WIPO Conversation on Intellectual Property and Frontier Technologies (Fourth Session),” Sep. 2019. Accessed: Apr. 14, 2022. [Online]. Available: https://www.wipo.int/export/sites/www/about-ip/en/frontier_technologies/interventions/pdf/ind_li.pdf

[28] S. Safavi, “UC Irvine UC Irvine Electronic Theses and Dissertations Title Novel detection, optimization, and monitoring techniques for neurological disorders,” 2020. Accessed: Apr. 14, 2022. [Online]. Available: <https://escholarship.org/content/qt6078571f/qt6078571f.pdf?t=qcsc0h>

[29] H. Wang, K. Tian, Z. Wu, and L. Wang, “A Short Text Classification Method Based on Convolutional Neural Network and Semantic Extension,” *International Journal of Computational Intelligence Systems*, vol. 14, no. 1, p. 367, 2020, doi: 10.2991/ijcis.d.201207.001.

[30] B. Ramzan *et al.*, “An Intelligent Data Analysis for Recommendation Systems Using Machine Learning,” *Scientific Programming*, vol. 2019, pp. 1–20, Oct. 2019, doi: 10.1155/2019/5941096.

[31] The Star, “Malaysia is 2nd in Asia for youth cyberbullying,” *The Star*, Jan. 14, 2022. <https://www.thestar.com.my/news/nation/2022/01/14/malaysia-is-2nd-in-asia-for-youth-cyberbullying>

APPENDICES

APPENDIX A: WEEKLY LOG

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: S3, Y3	Study week no.: Week 1 & Week 2
Student Name & ID: Yeong Su Yen 18ACB01410	
Supervisor: Dr Tong Dong Ling	
Project Title: Cyberbullying Detection: A Machine Learning Approach	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Make changes to the introduction of the report
- Create Gantt Chart for FYP2

2. WORK TO BE DONE

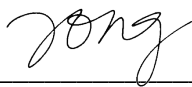
- Find examples of coding of implementing Bag of Words model and TF-IDF
- Make changes to the Gantt Chart

3. PROBLEMS ENCOUNTERED

- The details of the Gantt Chart is not specific and need to make changes to it

4. SELF EVALUATION OF THE PROGRESS

- I will plan my project tasks properly.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: S3, Y3	Study week no.: Week 3 & Week 4
Student Name & ID: Yeong Su Yen 18ACB01410	
Supervisor: Dr Tong Dong Ling	
Project Title: Cyberbullying Detection: A Machine Learning Approach	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Found examples of coding of implementing Bag of Words model and TF-IDF
- Wrote code to compare both methods to see which is better
- Made changes to the Gantt Chart

2. WORK TO BE DONE

- Find out how to build a corpus from the existing dataset

3. PROBLEMS ENCOUNTERED

- The code to transform word to numerical form have error but managed to fix it

4. SELF EVALUATION OF THE PROGRESS

- I will try to understand the code and won't just copy it without understanding it



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: S3, Y3	Study week no.: Week 5 & Week 6
Student Name & ID: Yeong Su Yen 18ACB01410	
Supervisor: Dr Tong Dong Ling	
Project Title: Cyberbullying Detection: A Machine Learning Approach	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Found out how to build a corpus from the existing dataset
- Wrote code to extract the words from the existing dataset

2. WORK TO BE DONE

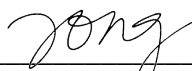
- Create wireframes for new web pages, such as Blog page and Statistic page
- Learn how to use Flask framework
- Start to write the code for Home page

3. PROBLEMS ENCOUNTERED

- The method VADER uses a long time to extract all abusive words but it is done and all words are extracted out

4. SELF EVALUATION OF THE PROGRESS

- I will try to understand the code and won't just copy it without understanding it



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: S3, Y3	Study week no.: Week 7 & Week 8
Student Name & ID: Yeong Su Yen 18ACB01410	
Supervisor: Dr Tong Dong Ling	
Project Title: Cyberbullying Detection: A Machine Learning Approach	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Created wireframes for new web pages, such as Blog page and Statistic page
- Learned how to use Flask framework
- Wrote the code for Home page (a draft)

2. WORK TO BE DONE

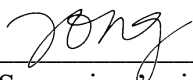
- Draw use case diagram, entity relationship diagram and architecture diagram
- Write a separate program to train the model with different machine learning algorithm to pick the one with highest accuracy

3. PROBLEMS ENCOUNTERED

- Need to make changes to the wireframe because some of the information are missing

4. SELF EVALUATION OF THE PROGRESS

- I need to think and find out what are my end deliverable of this project



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: S3, Y3	Study week no.: Week 9 & Week 10
Student Name & ID: Yeong Su Yen 18ACB01410	
Supervisor: Dr Tong Dong Ling	
Project Title: Cyberbullying Detection: A Machine Learning Approach	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Drew use case diagram, entity relationship diagram and architecture diagram
- Wrote a separate program to train the model with different machine learning algorithm to pick the one with highest accuracy

2. WORK TO BE DONE

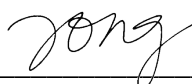
- Write use case description for each use case
- Try model validation to see how the classifiers perform from different set of test sets

3. PROBLEMS ENCOUNTERED

- Accuracy of the model in training dataset is high, but did not perform well on testing dataset

4. SELF EVALUATION OF THE PROGRESS

- I need to learn how to analyze the results after training the model



Supervisor's signature



Student's signature

APPENDIX B: CODES

These were the build the cyberbullying classifier with Bag of Words Model and Support Vector Machine.

```
In [38]: # change text to numerical form with BOW

# BOW
print("Bag of Words.....")
my_stopWords = stop_words
count_vectorizer = CountVectorizer(stop_words= my_stopWords)
bow_vector_final = count_vectorizer.fit(x_train)
bow_x_train_final = count_vectorizer.transform(x_train)

bow_x_test_final = count_vectorizer.transform(x_test)

Bag of Words.....

In [43]: # train the model and test the model

final_model = svm.SVC(C=0.1, kernel='linear', random_state = 42, probability=True)
final_model = sklearn.base.clone(final_model)

final_model.fit(bow_x_train_final, y_train_label)
final_model_predicted = final_model.predict(bow_x_test_final)

In [44]: # print classification report
print("Support Vector Machine")
print(accuracy_score(y_test_label, final_model_predicted))
print(classification_report(y_test_label, final_model_predicted))

# print confusion matrix
final_cm = confusion_matrix(y_test_label, final_model_predicted)
sns.heatmap(final_cm, annot=True, fmt = '.2f')
plt.title('Confusion Matrix for Final Model (SVM + BOW)')
plt.ylabel('Actual Values')
plt.xlabel('Predicted Values')
plt.show()

In [46]: # build pipeline to save model

pipeline = Pipeline([['bow', CountVectorizer(stop_words= my_stopWords)],
 ('clf', svm.SVC(C=0.1, kernel='linear', random_state = 42, probability=True))])

model = pipeline.fit(x_train, y_train_label)

In [123]: # Look at the accuracy of the model after pipeline
accuracy = model.score(x_test, y_test_label)
accuracy2 = accuracy_score(y_test_label, final_model_predicted)

In [124]: print('Accuracy score for pipeline model: ', accuracy)
print('Accuracy score for model before pipeline: ', accuracy2)

Accuracy score for pipeline model:  0.9261119081779053
Accuracy score for model before pipeline:  0.9261119081779053

In [121]: # dump the pipeline model
# Load this model into web application for classification
dump(pipeline, filename="text_classification.joblib")

Out[121]: ['text_classification.joblib']
```

These were the codes to build a corpus for abusive word. A list was created to store all the words found.

```
#find the polarity of each word and put it in a List
abusiveWords2 = [] #create a List
positiveWords2 = []
neutralWords2 = []
i = len(testData2['tokenized'])
j = 0

while i != 0:
    for x in testData2['tokenized'][j]:
        #print(x)
        word = x
        polarity = getPolarity_TB(word)
        #print('Word: {} Polarity: {}'.format(word, polarity))
        if polarity <= 0.0:
            abusiveWords2.append(word)
        elif polarity == 0.0:
            neutralWords2.append(word)
        else:
            positiveWords2.append(word)
    i = i - 1
    j = j + 1
```

This was used to build the complete set of abusive words.

```
import re
import string
from nltk.corpus.reader import WordListCorpusReader

w = WordListCorpusReader('.', ['C:\\NLP (fyp2)\\nltk_data\\corpora\\dataset\\AbusiveWords-ver1.txt'])
wordList = w.words()

wordString = " "

wordString = wordString.join(wordList)

# convert all words to capitalize Letters

caps = wordString.title()
caps_list = list(caps.split(" "))

#copy all capitalize words into a txt file
with open('capsList.txt', 'w') as fp:
    # write each item on a new line
    for item in caps_list:
        # write each item on a new line
        fp.write("%s\n" % item)
    print('Done')

# convert all words to uppercase Letters

upper = wordString.upper()
upper_list = list(upper.split(" "))

#copy all uppercase words into a txt file
with open('upperList.txt', 'w') as fp:
    # write each item on a new line
    for item in upper_list:
        # write each item on a new line
        fp.write("%s\n" % item)
    print('Done')
```

These were the codes to find likelihood of the message being a cyberbullying post. It is a function defined in the flask application.

```
# function to find likelihood
def findLikelihood(input):

    proba = pipeline.predict_proba([input])[:,1]
    pred = pipeline.predict([input])
    percentage_proba = proba*100

    return pred, percentage_proba
```

It is a snippet of code on how to categorize the message into different group of likelihood.

```
# predict likelihood
pred, percentage_proba = findLikelihood(msg)

if (pred == 1) & (percentage_proba >= 90) & (percentage_proba <= 100):
    extremely_high = "Likelihood of cyberbullying is extremely high"

    return render_template("index.html", cForm=cForm, likelihood=extremely_high,
                           cyberbullying=cyberbullying, abusive_word=abusive_word)

elif (pred == 1) & (percentage_proba >= 70) & (percentage_proba < 90):
    very_high = "Likelihood of cyberbullying is very high"
    return render_template("index.html", cForm=cForm, likelihood=very_high,
                           cyberbullying=cyberbullying, abusive_word=abusive_word )

elif (pred == 1) & (percentage_proba >= 50) & (percentage_proba < 70):
    high = "Likelihood of cyberbullying is high"
    return render_template("index.html", cForm=cForm, likelihood=high,
                           cyberbullying=cyberbullying, abusive_word=abusive_word)
```


These were the codes to find the abusive words in a sentence. It is a function defined in the flask application.

```
# function to find abusive word
def wordSearch(t):

    wordList = [] #list of words from corpus
    wordFound = [] #list of abusive words extracted from the sentence

    #read the corpus
    try:
        w = WordListCorpusReader('.', ['nltk_data\corpora\dataset\AbusiveWords (final).txt'])
        wordList = w.words()
    except:
        print("File is not found!")

    #user input
    text = " "
    text = t

    #split the sentence into individual word
    result = re.sub('[+string.punctuation+]', '', text).split()

    for i in wordList:
        for j in result:
            if i == j:
                wordFound.append(j)

    return wordFound
```

```
# find abusive word
abusiveWord_list= wordSearch(msg)
abusive_word= abusive_word.join(abusiveWord_list)

if len(abusiveWord_list) == 0:
    abusive_word = "There is no abusive word!"
```

This is written in the function that would map the results to the specified url.

POSTER

Faculty of Information and Communication
Technology



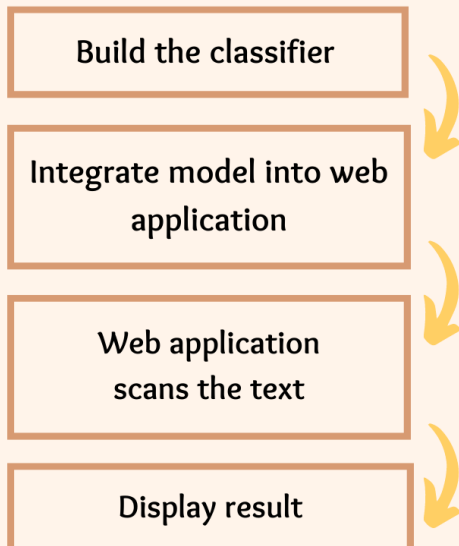
CYBERBULLYING DETECTION USING SVM AND BOW



INTRODUCTION

Cyberbullying cases are increasing as more people are using the Internet. This project focuses on creating a classifier that can detect cyberbullying text correctly. A web app was built to simulate the cyberbullying activities that might happen in a social media site.

HOW IT IS BUILT



OBJECTIVES

- To find a machine learning algorithm that can correctly predict a cyberbullying text.
- To detect abusive words in a message.

CONCLUSION

- The classifier has an accuracy of 93% and it has a good precision score.
- The classifier is able to detect cyberbullying activities in the web application.

Student: Yeong Su Yen
Bachelor of Computer Science (Honours)

Project Supervisor : Dr. Tong Dong Ling

PLAGIARISM CHECK RESULT

fyp2 all chapters ver 2

ORIGINALITY REPORT

4%	2%	1%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universiti Tunku Abdul Rahman Student Paper	2%
2	eprints.utar.edu.my Internet Source	<1%
3	Submitted to Nanyang Polytechnic Student Paper	<1%
4	Ahthasham Sajid, Muhammad Awais, Mirza Amir Mehmood, Shazia Batool, Amir Shahzad, Afia Zafar. "Patient's Feedback Platform for Quality of Services via "Free Text Analysis" in Healthcare Industry", EMITTER International Journal of Engineering Technology, 2020 Publication	<1%
5	Submitted to Universiti Teknologi Malaysia Student Paper	<1%
6	Submitted to Xiamen University Student Paper	<1%
7	Marina Binti Muhamad, Fathiah Binti Mohamed Zuki. "Attitude and Perception on The Disposal of Pharmaceuticals and Personal	<1%

Care Products In Malaysia: A Pilot Study", IOP
Conference Series: Materials Science and
Engineering, 2020
Publication

8	jurnal.idu.ac.id Internet Source	<1 %
9	Submitted to National Institute of Technology, Rourkela Student Paper	<1 %
10	Submitted to University Of Tasmania Student Paper	<1 %
11	www.researchgate.net Internet Source	<1 %
12	The Definitive Guide to Drupal 7, 2011. Publication	<1 %
13	e-string.com Internet Source	<1 %
14	etd.astu.edu.et Internet Source	<1 %
15	www.ncbi.nlm.nih.gov Internet Source	<1 %

Exclude quotes On
Exclude bibliography On

Exclude matches < 8 words

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1 of 1




FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Yeong Su Yen
ID Number(s)	18ACB01410
Programme / Course	Bachelor of Computer Science
Title of Final Year Project	Cyberbullying Detection: A Machine Learning Approach

Similarity	Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)
Overall similarity index: <u>4</u> % Similarity by source Internet Sources: <u>2</u> % Publications: <u>1</u> % Student Papers: <u>3</u> %	
Number of individual sources listed of more than 3% similarity: <u>0</u>	
Parameters of originality required and limits approved by UTAR are as Follows: (i) Overall similarity index is 20% and below, and (ii) Matching of individual sources listed must be less than 3% each, and (iii) Matching texts in continuous block must not exceed 8 words <i>Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.</i>	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.



 Signature of Supervisor

Name: Dr Tong Dong Ling

Date: 9 September 2022

 Signature of Co-Supervisor

Name: _____

Date: _____



UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	18ACB01410
Student Name	Yeong Su Yen
Supervisor Name	Dr Tong Dong Ling

TICK (✓)	DOCUMENT ITEMS
	Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
	Front Plastic Cover (for hardcopy)
✓	Title Page
✓	Signed Report Status Declaration Form
✓	Signed FYP Thesis Submission Form
✓	Signed form of the Declaration of Originality
✓	Acknowledgement
✓	Abstract
✓	Table of Contents
✓	List of Figures (if applicable)
✓	List of Tables (if applicable)
	List of Symbols (if applicable)
✓	List of Abbreviations (if applicable)
✓	Chapters / Content
✓	Bibliography (or References)
✓	All references in bibliography are cited in the thesis, especially in the chapter of literature review
✓	Appendices (if applicable)
✓	Weekly Log
✓	Poster
✓	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
✓	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

(Signature of Student)

Date: 9 September 2022

