SUICIDE IDEATION DETECTION AND RESPONSE SYSTEM FOR TEXTUAL SOCIAL MEDIA POSTS

LIM YAN QIAN

UNIVERSITI TUNKU ABDUL RAHMAN

SUICIDE IDEATION DETECTION AND RESPONSE SYSTEM FOR TEXTUAL SOCIAL MEDIA POSTS

LIM YAN QIAN

A project report submitted in partial fulfilment of the requirements for the award of Bachelor of Science (Honours) Software Engineering

Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

September 2022

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : *AmeliaYqian*

Name : Lim Yan Qian

ID No. : 2005778

Date : 26 September 2022

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"SUICIDE IDEATION DETECTION AND RESPONSE SYSTEM FOR TEXTUAL SOCIAL MEDIA POSTS"** was prepared by **LIM YAN QIAN** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Science (Hons) Software Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature : 

Supervisor : Ts. Dr Loo Yim Ling

Date : 26 September 2022

Signature :

Co-Supervisor : - NIL -

Date : - NIL -

ACKNOWLEDGEMENTS

I would like to extend my sincere appreciation towards my supervisor, Dr. Loo Yim Ling for her patience and guidance throughout the development of this project. Her professionalism and advice have always kept me motivated to put in more effort to go above and beyond the project's deliverables, as well as my personal goals for this project. Thank you for showing me the power in believing in my work.

Most importantly, I am grateful for my dearest family members and close friends that have always encouraged, supported and inspired me to achieve my full potential through this project. With their kind words and support, I was able to stay committed and resilient throughout all the challenges that I have faced in a supportive environment, which contributed even more value in this work.

Lastly, I am also thankful for the academia/ online tech community for allowing me to gain a deeper understanding and broader perspective on various technological developments and the most effective way to utilise it for problem solving. From there, I was able to refine my technical skills and articulate my discoveries with greater depth and breadth of understanding.

ABSTRACT

Suicide is the fourth most common cause of mortality among youths. With the emergence of digital technology, individuals are increasingly using social media platforms as a "safe space" to express their suicidal tendencies. As such, this project focuses on developing a comprehensive web application that supports real-time classification of tweets into three suicide risk categories and triggers a tailored crisis response that targets specific suicide risk levels. To that end, this project covers the end-to-end activities from model development, which uses Natural Language Processing and feature extraction techniques to improve the model's classification performance, to its deployment on Flask web application framework for real-time monitoring and detection of tweets, and finally the initiation of proactive responses tailored to specific suicide risk levels. The methodology adopted for this project is Scrum methodology, which runs on 3 sprints. Results showed that the approach used significantly improved the classification performance when benchmarked with existing works. The model was integrated into the web system developed in this project and tested with random tweet samples of varying suicide risk. Overall, it was shown that the model maintained high performance results and the system was able to proactively trigger the correct response that addressed each suicide risk, which proves the efficacy of the model in real-time environment.

TABLE OF CONTENTS

DECLARATION **i**
APPROVAL FOR SUBMISSION **ii**
ACKNOWLEDGEMENTS **iv**
ABSTRACT **v**
TABLE OF CONTENTS **vi**
LIST OF TABLES **xi**
LIST OF FIGURES **xii**
LIST OF SYMBOLS / ABBREVIATIONS **xvi**
LIST OF APPENDICES **xviii**

CHAPTER

1 INTRODUCTION **1**
1.1 General Introduction 1
1.2 Importance of the Study 2
1.3 Problem Statement 4
1.3.1 Misuse of Social Media Platforms 4
1.3.2 Error and Inefficiencies of the Current Response System 4
1.3.3 Data and Privacy Issues 5
1.4 Aim and Objectives 6
1.5 Project Solution 7
1.6 Proposed Approach 8
1.6.1 Model Development 8
1.6.2 Web Application Development 8
1.6.3 System Evaluation 8
1.7 Scope and Limitation of the Study 9
1.7.1 User Demographic 9
1.7.2 Content Used 9
1.7.3 Features Included 10

	1.8	Contribution of the Study	11
		1.8.1 Analysis on Findings	11
		1.8.2 Improved Model	11
		1.8.3 Validated Detection and Response Mechanism	12
	1.9	Outline of the Report	12
2	**LITERATURE REVIEW**		**13**
	2.1	Introduction	13
	2.2	Data Sources and Collection Methods	18
		2.2.1 Twitter	19
		2.2.2 Reddit	20
		2.2.3 Facebook	21
		2.2.4 Summary	21
	2.3	Natural Language Processing Method and Tools	22
		2.3.1 Feature Extraction	23
		2.3.2 NLP Tools	27
	2.4	Machine Learning Model	31
		2.4.1 Random Forest	33
		2.4.2 Support Vector Machine (SVM)	34
		2.4.3 Logistic Regression (LR)	34
		2.4.4 Summary	35
	2.5	Existing Works	35
		2.5.1 Rabani, Khan and Khanday (2020)	36
		2.5.2 Ji et al. (2018)	36
		2.5.3 Liu et al. (2019)	37
		2.5.4 Yang et al. (2021)	37
		2.5.5 Gomes de Andrade et al. (2018)	38
		2.5.6 Summary	40
	2.6	Software Engineering Methodologies	41
		2.6.1 Waterfall	42
		2.6.2 Rapid Application Development (RAD)	43
		2.6.3 Scrum	44
		2.6.4 Summary	45
	2.7	Web Application Framework	46

2.7.1 React 46
2.7.2 Laravel 47
2.7.3 Flask 47
2.7.4 Summary 47
3 METHODOLOGY AND WORK PLAN 49
3.1 Introduction 49
3.2 Software Development Methodology 49
3.2.1 Initiation 50
3.2.2 Planning 51
3.2.3 Design 51
3.2.4 Development 52
3.2.5 Testing 53
3.2.6 Close-off 53
3.3 Development Tools 53
3.3.1 Twitter API 53
3.3.2 Jupyter Notebook 54
3.3.3 Flask 55
3.4 Work Plan 56
3.4.1 Work Breakdown Structure (WBS) 56
3.4.2 Gantt Chart 59
4 PROJECT SPECIFICATIONS 61
4.1 Introduction 61
4.2 Machine Learning Model 61
4.2.1 Data Collection and Preparation 62
4.2.2 Data Pre-processing 64
4.2.3 Feature Extraction 64
4.2.4 Model Training 66
4.2.5 Performance Evaluation 66
4.3 System Requirements 68
4.3.1 Functional Requirements 68
4.3.2 Non-functional Requirements 68
4.4 Use Case 69
4.4.1 Use Case Diagram 69
4.4.2 Use Case Description 70

	4.5	Prototype Design	74
		4.5.1 Flowchart	74
		4.5.2 Web Application Prototype	75
5	**DATA MODELLING**		**78**
	5.1	Data Collection	78
	5.2	Data Annotation	79
	5.3	Data Pre-processing	82
		5.3.1 Removal of Redundant Words and Characters	82
		5.3.2 Tokenization and Lemmatization	82
	5.4	Feature Extraction	83
		5.4.1 PoS Tagging	83
		5.4.2 Sentiment Analysis	85
		5.4.3 TF-IDF	88
	5.5	Model Training	91
6	**SYSTEM DEVELOPMENT AND MODEL INTEGRATION**		**92**
	6.1	Introduction	92
	6.2	Model Integration	92
	6.3	System Development	93
		6.3.1 Backend Development	93
		6.3.2 Frontend Development	95
	6.4	Summary	102
7	**RESULTS AND DISCUSSIONS**		**103**
	7.1	Introduction	103
	7.2	Model Performance Evaluation	103
		7.2.1 Performance Analysis	103
		7.2.2 Summary	107
	7.3	System Testing	107
		7.3.1 Set A	109
		7.3.2 Set B	113
	7.4	Summary	116
8	**CONCLUSIONS AND RECOMMENDATIONS**		**118**
	8.1	Conclusions	118

8.2 Recommendations for future work 120

REFERENCES **121**

APPENDICES **127**

LIST OF TABLES

Table 2.1:	Literature Review	13
Table 2.2:	Natural Language Processing Tasks	22
Table 2.3:	Machine Learning Models	31
Table 2.4:	Strengths and Limitations of Random Forest	33
Table 2.5:	Strengths and Limitations of SVM	34
Table 2.5:	Strengths and Limitations of LR	34
Table 2.6:	Comparison of Existing Works	35
Table 2.7:	Comparison of Software Development Methodologies	45
Table 3.1:	Python Libraries Used	54
Table 4.1:	Risk Classification	63
Table 5.1:	Excerpt of Tweets	80
Table 6.1:	HTTP Endpoints used	94
Table 7.1:	Performance Results from Existing Works (Multiclass Classification)	106
Table 7.2:	Performance Results from Existing Works (Binary Classification)	107

LIST OF FIGURES

Figure 1.1:	System Overview	7
Figure 2.1:	Waterfall Approach (Pfleeger and Atlee, 2006 citied in Adenowo and Adenowo, 2013, p.429)	42
Figure 2.2:	Overview on Scrum Methodology (Van Casteren, 2017)	44
Figure 3.1:	Overview on Scrum Methodology used	49
Figure 3.2:	Work Breakdown Structure Part 1	56
Figure 3.3:	Work Breakdown Structure Part 2	57
Figure 3.4:	Work Breakdown Structure Part 3	58
Figure 3.5:	Gantt Chart Part 1	59
Figure 3.6:	Gantt Chart Part 2	59
Figure 3.7:	Gantt Chart Part 3	59
Figure 3.8:	Gantt Chart Part 4	60
Figure 3.9:	Gantt Chart Part 5	60
Figure 4.1:	Overview on Machine Learning Model Development	61
Figure 4.2:	Overview on Data Collection and Preparation	62
Figure 4.3:	Suicide-indicative Terms (Parrott et al., 2020)	63
Figure 4.4:	Data Pre-processing Method	64
Figure 4.5:	Feature Extraction	65
Figure 4.6:	Use Case Diagram	69
Figure 4.7:	Flowchart for Web Application	74
Figure 4.8:	Web Application Prototype (Dashboard)	75
Figure 4.9:	Web Application Prototype (Pending Review Tab)	75
Figure 4.10:	Web Application Prototype (View All Tweets Tab)	75
Figure 4.11:	Web Application Prototype (Select Action Modal)	76

Figure 4.12:	Web Application Prototype (Update Action Modal)	76
Figure 4.13:	Web Application Prototype (“Safe to Ignore” Feedback Modal)	76
Figure 4.14:	Web Application Prototype (“Alert the Authorities” Feedback Modal)	77
Figure 5.1:	Twitter Authentication Code Snippet	78
Figure 5.2:	Data Collection Code Snippet	79
Figure 5.3:	Top 20 Commonly Used Words in Medium Suicide Risk Level	80
Figure 5.4:	Top 20 Commonly Used Words used in High Suicide Risk Level	81
Figure 5.5:	Standardization of Contractions	82
Figure 5.6:	Snippet of Tokenized and Lemmatized Tweets	82
Figure 5.7:	PoS Tags on Sample Tweet	83
Figure 5.8:	PoS Tag Feature Set for Sample Tweet	84
Figure 5.9:	Top 10 PoS Tags for Medium Risk Tweets	84
Figure 5.10:	Top 10 PoS Tags for High Risk Tweets	85
Figure 5.11:	VADER Score for Sample Tweet with Low Suicide Risk	86
Figure 5.12:	VADER Score for Sample Tweet with Medium Suicide Risk	86
Figure 5.13:	VADER Score for Sample Tweet with Medium Suicide Risk	87
Figure 5.14:	VADER Score for Sample Tweet with High Suicide Risk	87
Figure 5.15:	VADER Score for Sample Tweet with High Suicide Risk	88
Figure 5.16:	Top 25 Terms with Highest Mean TF-IDF score for Medium Suicide Risk	89
Figure 5.17:	Top 25 Terms with Highest Mean TF-IDF Score for High Suicide Risk Level	90
Figure 5.18:	Model Training Code Snippet	91

Figure 6.1:	Overview of System Development	92
Figure 6.2:	Model Serialization Code Snippet	93
Figure 6.3:	Model Deserialization Code Snippet	93
Figure 6.4:	Model Prediction Code Snippet	93
Figure 6.5:	Overview on Backend Architecture for POST Request	94
Figure 6.6:	View Dashboard	95
Figure 6.7:	View Dashboard (View All Tweets Tab)	96
Figure 6.8:	View Real-Time Statistics	96
Figure 6.9:	Select Action Button	97
Figure 6.10:	Select Action Modal	98
Figure 6.11:	Response for Medium Suicide Risk	98
Figure 6.12:	Response for High Suicide Risk	98
Figure 6.13:	Update Action	99
Figure 6.14:	Update Action Modal	99
Figure 6.15:	Create Tweet	100
Figure 6.16:	Feedback Modal for Low Suicide Risk	101
Figure 6.17:	Feedback Modal for Medium Suicide Risk	101
Figure 6.18:	Feedback Modal for High Suicide Risk	102
Figure 7.1:	Confusion Matrix	104
Figure 7.2:	Histograms of Sentiment Score by Suicide Risk Level	105
Figure 7.3:	Overview on Test Samples Used	108
Figure 7.4:	Test Samples for Low Suicide Risk	109
Figure 7.5:	Additional Sample	110
Figure 7.6:	Test Samples for Medium Suicide Risk	110
Figure 7.7:	Test Samples for High Suicide Risk	111

Figure 7.8: Confusion Matrix for Set A 112

Figure 7.9: Confusion Matrix for Set B 113

Figure 7.10: Samples of Correctly Classified Tweets 114

Figure 7.11: Samples of Misclassified Tweets 114

Figure 7.12: Samples of Misclassified Tweets 115

Appendix A: Graphs 127

Appendix B: Tables 129

LIST OF SYMBOLS / ABBREVIATIONS

t	term
AI	Artificial Intelligence
API	Application Programming Interface
BoW	Bag of Words
CNN	Convolutional Neural Network
FN	False Negative
FP	False Positive
HCI	Human Computer Interaction
IDF	Inverse Document Frequency
JSON	JavaScript Object Notation
LIWC	Linguistic Inquiry and Word Count
LR	Logistic Regression
LSTM	Long Short-Term Memory
ML	Machine Learning
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
PoS	Part-of-Speech
RAD	Rapid Application Development
RF	Random Forest
REST/RESTful	Representational State Transfer
SCLIWC	Simplified Chinese Linguistic Inquiry and Word Count
SDLC	Software Development Lifecycle
SVM	Support Vector Machine
TF	Term Frequency
TF-IDF	Term Frequency-Inverse Document Frequency
TN	True Negative
TP	True Positive
UI	User Interface
UN	United Nations
URL	Uniform Resource Link
VADER	Valence Aware Dictionary and Sentiment Reasoner

WBS	Work Breakdown Structure
WHO	World Health Organization

LIST OF APPENDICES

Appendix A: Graphs 127

Appendix B: Tables 129

CHAPTER 1

INTRODUCTION

1.1 General Introduction

Communication is the essence to express one's thoughts, feelings, desires or mental states. Often times, people utilize social media applications such as Facebook, Reddit and Twitter as a medium to communicate and connect with others. A massive increase in the accessibility and ease of use of social media platforms have resulted in the ability of a social media user's communication content to reach a wider audience. Thus, making social media applications a fundamental medium of communication for people and organizations all around the world to share information as well as their ideas. This significance has resulted in an increase of individuals who turn to social media platforms as a "safe space" for themselves to express their feelings and suicide tendencies (Ji et al., 2021), in hopes of expressing their actual feelings without being judged, as well as connecting with others who are in similar situations. Consequently, Internet sources and social media applications have provided an avenue for suicidal content to be effortlessly accessible and increasingly popularised. In fact, according to a study carried out by Dunlop (2011) on 719 people, 59% of those surveyed stated that they learned about suicidal topics through online sources.

For this reason, a crucial step towards suicide prevention is through early intervention by connecting the individual at risk of suicide to reliable crisis and mental health resources (Mann et al, 2005 cited in Hassan, Hassan and Zakia, 2020). The series of events that occurs in between experiencing suicidal tendencies and carrying out the suicide attempt has chances to take place within a few hours (Millner, Lee and Nock, 2016). Therefore, this serves as a critical, real-time window of opportunity to intervene and bring the individual back into their health and safety.

However, communication blockers may exist in which the recipient fails to recognize or dismisses the responsibility of reporting a suicide attempt due to the social stigma of mental health issues, or even out of fear of

committing a legal crime (Soron, 2019). In addition to that, existing social media applications have little to no proper policies or mechanism in place to identify these warning signs. Even with a mechanism in place, many users fail to understand the risk and take the necessary action to mitigate the issue at an earlier stage (Rabani, Khan and Khanday, 2020).

This suggests a need for a proper mechanism in place to identify and initiate the proper response to assist individuals who expressed suicide ideation through social media posts, as it is possible that this individual's life can be saved. Addressing these calls, this project focuses on developing a comprehensive web application that detects suicidal tendencies based on the user's social media posts and triggers a tailored response once such ideation is detected.

1.2 Importance of the Study

The Sustainable Development Goals adopted by the United Nations (UN) have listed mental health and overall well-being as one of its 17 critical agendas, identifying it as a key priority towards the sustainable development of humankind. Despite that, mental health disorders such as anxiety disorder and depression are becoming increasingly prevalent across many countries. In fact, suicide-related deaths were strongly linked to existing mental illnesses, with statistics showing that 90% of suicide-related deaths in high-income nations were linked to mental health issues (World Health Organization, 2018). Moreover, worldwide statistics obtained from the World Health Organization (WHO) shows that over 700,000 people commit suicide every year, making it the fourth main cause of mortality among 15–29-year-olds in 2019 (World Health Organization, 2021).

The emergence of digital technology has accompanied this gradual increase in suicide-related deaths, with social media having grown pervasive and profoundly integrated in our lives during the last two decades (Soron and Shariful Islam, 2020). As such, communicating distress on social media has also become prevalent form used by individuals who are at risk of attempting suicide or any related forms of self harm (Soron and Shariful Islam, 2020). According to studies conducted by Pourmand et al. (2019), it was revealed that youths are more eager to communicate their suicidal thoughts via social media

applications such as Facebook and Twitter, although they are hesitant to do so during their scheduled medical visits with a mental health professional.

Besides that, stigma associated with mental health, as well as the lack of access to mental healthcare remain key challenges that deter one to seek help and treatment from healthcare professionals. According to studies by Soron (2019), it was found that the restricted access in communicating one's emotional suffering and seeking help from the current health and social system may have an influence on suicide. For instance, according to the law in countries such as Bangladesh, the act of attempting and encouraging suicide is classified as a criminal offense. As such, people in Bangladesh are more inclined to share their suicide ideation and plans through social media as they are afraid to do so openly in public for fear of being prosecuted. This phenomenon has led to the hidden at-risk individuals within the online community to turn to social media to discuss their suicidal thoughts as a way for them to receive social support and access professional help (Soron, 2019).

Furthermore, earlier attempts to administer suicide prevention and detection tools among social media platforms have sparked public debate about the practice of an individual's rights to freedom of speech and expression within the platform (Luxton, June and Fairall, 2012). Regardless of that, efforts to safeguard the safety and mental well-being of vulnerable end-users should remain a top priority and embraced across all mass media platforms.

In order to establish a balance between social media use and suicide prevention, a modern, data driven approach should be considered in order to tackle these issues, by leveraging on emerging technological trends such as Machine Leaning to step in and create safer spaces on social media applications. Compared to in-person conversations, a suicide ideation detection and response system could be more effective and efficient in identifying and responding to these red flags and subtle signs of suicidal thoughts that are expressed through the individual's social media posts.

1.3 Problem Statement

1.3.1 Misuse of Social Media Platforms

As social media exists as a prevalent part of our daily lives, companies tend to introduce new features on their platform from time to time for their users to explore and utilise. Apart from sharing and reacting to content that are shared on the different social networking sites, there are many interactive tools such as the polling feature in Instagram and Twitter that allow the users to create live polls that can be answered by their platform connections. However, there were incidents of social media misuse where individuals leveraged the openness of the social media platforms to gather support for their suicidal actions. For instance, an online user committed suicide after most respondents voted for "Death" in her Instagram poll (Fullerton, 2019). This incident raised discussion as the public questioned whether the "Death" respondents could face legal consequences for encouraging suicide.

Besides that, there are also cases where individuals who have committed suicide have been idolized in online forums, leading to heightened peer pressure to commit suicide (Luxton, June and Fairall, 2012). Interactions within such discussion forums and private chats also comes in reverse effect as it glamourises the act of suicide and reduces the individual's doubts and fears towards it (Luxton, June and Fairall, 2012). This would then further influence the judgement of those individuals at risk due to the vulnerability of their emotional state. Although there is no way to control the exponential growth of social networking sites over the recent years, it is imperative to ensure that extra consideration is given towards formulating and introducing a suicide prevention response that curbs the misuse of social media platform that influences pro suicide behaviour.

1.3.2 Error and Inefficiencies of the Current Response System

Following the growing number of suicide cases that are related to social media applications, major social media companies have taken a proactive approach to address the issue. As such, Facebook have rolled out its first machine learning powered suicide detection system in 2017, which aims to identify potential suicide or self-harm content (Roshen, 2017). Thereafter, companies such as

Instagram have also introduced similar responses. Although there exists a suicide response and alert system in such social media applications, the current system is still inefficient in providing the appropriate response to address the issue. For instance, a Twitter user was once locked out of her account after being mistakenly reported for publishing suicidal content (Vaughn, 2022). Furthermore, the links that was supposed to direct her to mental health resources were not functioning properly (Vaughn, 2022). This shows the inefficiency and lack of integrity in the existing mechanism used by social media applications to positively identify possible threats of suicide and provide credible response for individuals who are in need of such assistance.

1.3.3 Data and Privacy Issues

The utilization of social media applications can be a huge leap in bridging the gap between suicide prevention responses and the individual at risk. However, clinicians and researchers face a considerable amount of ethical and methodological difficulties when it comes to privacy in the digital age (Pourmand et al., 2019). The complexities surrounding the legal concerns when accessing, monitoring and filtering the online content have been a subject of public debate. For instance, in recognition of the dire need of an effective suicide alert system in the market, the Samaritans organization previously released a Twitter plug-in, titled “Samaritan’s Radar” that aims to detect and alert the user if any words or phrases indicating the need for immediate intervention were detected among their followed user accounts (Resnik et al., 2020). The main aim of this application is to allow for users to extend their support to the user that was being flagged as “at risk” by the algorithm (Coppersmith et al., 2018). However, the system was ultimately shut down shortly after its launch due to the severe criticism from the public on the privacy issues associated with the plug-in. The 2 primary concerns surrounding the system was that: 1) Users were not aware that their content was being analysed; 2) Fear that this information would be misused for malicious intentions such as cyberbullying (Coppersmith et al., 2018). Hence, this suggests a need for further research to look into coordinating efforts with consideration for the ethical, legal and social implication associated with the

use of machine learning in utilising publicly available data from online sources such as Facebook, Twitter and Reddit.

1.4 Aim and Objectives

To address the problems highlighted above, this project focuses on developing a web application for individuals at risk of suicide who demonstrate the need to connect with credible mental health resources or require immediate attention from their local authorities. Essentially, the primary aim of this web application is to create a pathway for timely and proactive crisis intervention that connect these at-risk individuals to professional help. As such, the 3 main objectives of this project are:

1. To develop a Random Forest model and apply Natural Language Processing (NLP) techniques to improve the classification performance of detecting suicide ideation among textual posts on Twitter
2. To develop a web application that integrates Random Forest model to enable real-time classification of tweets into its associated level of suicide risk
3. To develop proactive and personalized distress response for tweets that are successfully detected with suicide risk level from medium to high and prioritize tweets with high suicide risk for further action by moderator

1.5 Project Solution

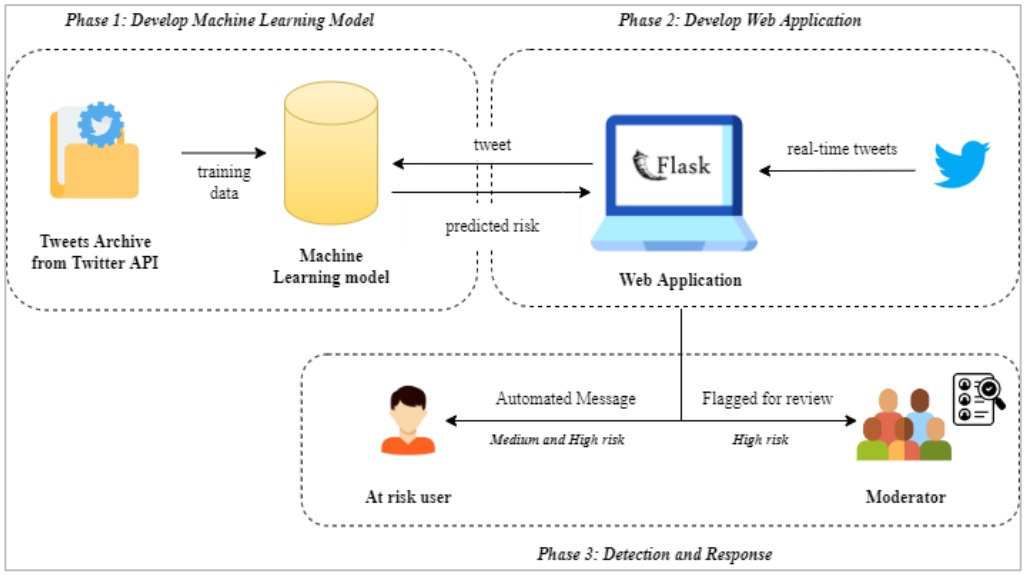


Figure 1.1: System Overview

Figure 1.1 depicts the three-phased platform that was developed through this project to effectively recognize and trigger the relevant response to address user's suicide ideation in textual social media posts. In the first phase, a combination of NLP techniques such as Term Frequency-Inverse Document Frequency, sentiment analysis and Part-of-Speech tagging was applied to form the feature sets from the dataset collected through the tweet archive from Twitter's Application Programming Interface (API). These feature sets represent the most significant information from the tweets which is then passed into the machine learning (ML) model, Random Forest for classification of the tweets into three suicide risk levels: 0: (Low Risk), 1: (Medium Risk) and 2: (High Risk). In the second phase, the trained model was integrated into the web application for real-time analysis of the tweets. The web application serves as a comprehensive platform for the content to be continuously monitored in order for suicide risk levels to be automatically labelled. In the last phase, the detection and response system was developed, such that if medium and high suicide risk is detected, an automated message will be sent to the at-risk user with links to mental health resources. Additionally, tweets detected with high suicide risk will be immediate flagged by the system. This is to prioritise these tweets and allow the moderator to

review the context of the tweet to determine whether it needs to be escalated to local authorities.

1.6 Proposed Approach

Scrum methodology was used for the development of this project, which mainly breaks down the system's features into 3 separate time-boxed sprints: model development, web application development and system evaluation. These sprints are logically related, each of which produces a defined output. This allow each major feature of the system to be developed incrementally and sequentially before integrating it to form the final and complete system.

1.6.1 Model Development

This sprint focuses on all tasks related to model development, which includes data collection, data annotation, data pre-processing, feature extraction and model training. The dataset was collected through Twitter API and annotated through human evaluation. NLP techniques were used to perform data pre-processing and feature extraction on the dataset. The feature set was then passed into the model to analyse the tweets and classify it according to its level of suicide risk.

1.6.2 Web Application Development

A Flask web application was built to integrate the trained model to detect real-time tweets automatically. The response system was configured and developed to send a direct message containing information and links to credible mental health resources to the author of the tweets that are classified with medium to high level of suicide risk by the system. This platform also provides a simple interface with basic functionalities for the moderator to review and determine the appropriate action to address tweets flagged with high suicide risk.

1.6.3 System Evaluation

The developed model was benchmarked with similar research studies to provide a basis to evaluate the model's performance. This is to evaluate if the approach used to train and build the classification model was able to

outperform the existing works, in terms of improving the classification performance. In addition to that, the web system was evaluated with random test samples to obtain an unbiased and fair evaluation of the system's performance on whether it could associate the tweet to its suicide risk as well as trigger the correct response accordingly.

1.7 Scope and Limitation of the Study

1.7.1 User Demographic

According to statistics obtained from Twitter, among the 206 million monetizable users worldwide, the United States is the leading country that accounts for approximately 77.7 million users on the platform as of October 2021 (Kemp, 2021). Further to that, research conducted in February 2021 found that 92% of the U.S adult users in the survey set their account to public, with the most active users by posting an average of 65 tweets every month (refer to Appendix A, Graph A-1 and Graph A-2) (McClain et al., 2022). Leveraging on the saturation of the content and accessibility of data that can be obtained, the target user group that this project will focus on is users located in the United States. This gives an opportunity for analysis to be conducted with a larger sample size, with an increased chance of gathering more quality data to assist the detection of tweets with suicide-related intent.

1.7.2 Content Used

The content that was used for the model training and real-time analysis will be strictly focused on textual posts that are in English language only. The main motivation for limiting the language of the content is because most of the content found on Twitter are written in English (Hong, Convertino and H. Chi, 2011). Therefore, such restriction was applied for this project, with hopes that it will help in increasing the quality of the data that is being trained through the model. This also ensures that the analysis of the model is not compromised during real-time analysis, as the data passed into the model is of the same language as its training data.

1.7.3 Features Included

For better representation of the system in a dynamic and intuitive manner, the web application developed through this project is a single page web application. This is also to allow the webpage to be continuously updated in real-time without requiring the user to initiate any prior user input or interaction with the application. With that, there are 3 main features that are included in the web application, which are: view dashboard, analyse tweet and flag tweets.

1.7.3.1 View Dashboard

This feature provides the moderators a general dashboard overview of the number of tweets that are pending for analysis by the moderator, as well as the total tweets that are detected by the system. The dashboard provides visualization charts which provides dynamic, high-level reporting on the trends and analysis of the real-time data. In addition to that, it also contains basic navigational features that helps to alert the moderator when there are incoming tweets that are pending their review. The overall purpose of this feature is to deliver a seamless user experience for the moderators to get a simple, clean and intuitive overview on the tweets that are captured and analysed by the model.

1.7.3.2 Analyse Tweet

This feature allows the moderator to evaluate the tweet with details such as its content, author, date, time and location when the tweet was sent. The main purpose of this feature is to supply the moderator sufficient and relevant information on the tweet to aid the decision making on whether the tweet under review requires further action and escalation to the local authorities. Hence, this feature is mainly used for tweets detected with high suicide risk, as the moderator may utilise the information to look into the Twitter profile and determine if the local authorities need to be contacted to further assist the individual at risk of suicide.

1.7.3.3 Flag Tweets

This feature allows the moderator to flag tweets that indicates that the individual is in imminent risk of suicide. The information collected from the tweet is utilized as a reference for the authorities to locate and identify the individual at risk of suicide. For example, if the moderator determines that the author of the tweet under review is on the verge of committing suicide, the tweet's GPS location and the phone number of the local authorities will be displayed on the web application as a reference for them to inform the local authorities. The purpose of this feature is for timely and proactive crisis intervention by providing support to these at-risk individuals through a higher authority.

1.8 Contribution of the Study

The present study reflects a comprehensive approach towards understanding how suicide detection, response and intervention can be effectively managed. Hence, the main contribution of the study can be separated into 3 different aspects:

1.8.1 Analysis on Findings

This study presents a detailed analysis on the findings obtained from the ML model, which identified the sentiment and linguistic characteristics associated with specific levels of suicide risk. These findings added to existing research as it provides an increased understanding on how these patterns aid the model in discerning different levels of suicide risk, which was not emphasised by previous studies of the same problem domain.

1.8.2 Improved Model

This study also provided a trained model for accurate detection of different levels of risk associated with suicide ideation within social media textual posts with an increased performance when benchmarked with related works. The methods and techniques used to develop the ML model were presented in detail to promote research transparency and provide an opportunity for further

exploration and extension in other research works under similar problem domain.

1.8.3 Validated Detection and Response Mechanism

Prior to this study, existing works often focused on either the detection or the response for suicidal social media content. Hence, this study contributed by presenting a validated approach towards streamlining the integration of suicide ideation detection and response under a single, fully functional web application. This includes the formulation of a proactive response that targets specific level of suicide risks. Hence, this provides an opportunity for other social media sites to seamlessly integrate this system into their platform.

1.9 Outline of the Report

The report is organized as follows: Chapter 2 details the literature review that is performed on existing works that are within the similar problem domain, as well as detailed research on the technical tools and techniques that are used in other studies. Chapter 3 shows the methodology and work plan deployed for the study. Chapter 4 describes the specific requirements of the project that are required to achieve the project objectives. Chapter 5 presents the approach and preliminary findings obtained from the data modelling process. Chapter 6 shows the integration of the model and the development of the complete system. Chapter 7 presents the discussion on the results and findings obtained through model training and system test. Finally, the conclusion and recommendations are presented in Chapter 8.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This section provides a detailed review on similar research works of the same problem domain. The following sub-sections are organized into 6 topics which includes: data sources and collection methods, NLP method and tools, machine learning model, existing approaches, software engineering methodologies and web application frameworks. Table 2.1 shows an overview on the publications included in the literature review.

Table 2.1: Literature Review

No	Publication	Objective	Origin and Size of Dataset	Strengths and Limitations
1	(Shing et al., 2018)	To analyse the level of suicide risk associated with online social media posts using different human assessment techniques and explore use of machine learning model to detect tweets with potential risk of suicide	Reddit; Final dataset consists of 865 user's post	**Strengths:** • High credibility and reliability of dataset, since it is manually annotated by a large team of mental health expert and nonexperts • Authors used multi-class classification **Limitations:** • Collection and analysis of data is from Reddit only

Table 2.1 (Continued)

2	(Ji et al, 2018)	To explore the early detection of suicide ideation through supervised learning classifiers	Reddit (more than 2549 posts) and Twitter (10288 tweets)	**Strengths:** • Combination of informative feature sets and effective classifier models for effective performance improvement • Large dataset used for training and testing **Limitations:** • Proposed approach implements single class classification only
3	(Tadesse et al., 2019)	To explore the application of deep learning architecture to build an effective machine learning model that detects suicidal posts on Reddit	Reddit; Final dataset contains 7201 posts	**Strengths:** • Adoption of hybrid framework for effective performance improvement • Large corpus used for training and testing **Limitations:** • Proposed approach implements single class classification only

Table 2.1 (Continued)

4	(Muhammad Shah et al., 2020)	To test the significance of applying feature extraction methods to create robust feature set in increasing the accuracy of suicide ideation detection	Reddit; Final dataset contains 7098 posts	**Strengths:** • Use of best feature extraction algorithms greatly reduced complexity of computation and significantly improved model performance • Large training corpus used **Limitations:** • Proposed approach implements single class classification only
5	(Rabani et.al., 2020)	To explore the feasibility of identifying tweets contain that suicide ideation using machine learning and ensemble methods	Twitter; Final dataset after data pre-processing had a total of 4266 tweets.	**Strengths:** • Use of an extensive combination of feature engineering techniques to improve model performance **Limitations:** • Proposed approach implements single class classification only
6	(O'Dea et al., 2015)	To test the feasibility of identifying the level of concern of tweets which make suicide-indicative references, regardless of whether it contains textual, audio or visual content	Twitter; Data passed into model was broken down into 2 sets: Set A (829 tweets), Set B (991 tweets)	**Strengths:** • Use of multi-class classification **Limitations:** • Limited size and range of suicide-related terms were included in the dataset

Table 2.1 (Continued)

7	(Mbarek et al., 2019)	Real-time identification of twitter profiles that contain account features and tweet content related to suicide using semantic analysis and machine learning	Twitter; Final dataset passed into model contains 785 posts	**Strengths:** • Development of unique feature set that includes features extracted from user's Twitter profile and their tweet content **Limitations:** • Proposed approach implements single class classification only
8	(Gomes de Andrade et al., 2018)	To explore the architecture of the suicide prevention feature used on Facebook and the ethical issues throughout the implementation of the tool	Facebook; N/A	**Strengths:** • Collaboration and validation from mental health organizations to analyse, curate and improve the implementation of the classifier **Limitations:** • This approach is only created, tested and used for Facebook platform
9	(Liu et al., 2019)	To develop and test the acceptability of combining proactive detection of those at risk of suicide with specialized crisis management	Weibo; Total comments analysed were 27007	**Strengths:** • Initiation of specialized crisis response that is tailored to each individual's needs **Limitations:** • Proposed approach implements single class classification • Requires voluntary participation from respondents in order to offer support services

Table 2.1 (Continued)

10	(Hassan et al., 2020)	To develop an automated conversational mechanism that integrates Human Computer Interaction (HCI) via virtual personal assistant device with NLP to detect suicide ideation and trigger response	Conversational speech captured by Google Home Mini; N/A	**Strengths:** • Proposed approach automatically connects users to mental health resources upon detection of suicide ideation **Limitations:** • Specification on how Dialogflow worked was not outlined • Since it is a pilot study, the proposed approach and trigger of the response was not tested on real world data
11	(Yang et al., 2021)	Review of "Tree Hole Action", a suicide screening and crisis response program that integrates advanced Artificial Intelligence algorithm and mental health services to monitor suicide risk in real time and provide practical crisis intervention	Weibo; N/A	**Strengths:** • Systematic, timely and effective approach used for suicide risk detection and crisis intervention that integrates online and offline support to enhance proactive intervention • Proposed approach supports multi-class classification **Limitations:** • Difficult to provide crisis support for individuals reluctant to participate during support

Table 2.1 (Continued)

12	(Nobles at al., 2018)	To develop a machine learning model that identifies period of suicidality among young adults using text message analysis	Text messages, based on voluntary participation	**Strengths:** • Language changes were detected one an individual shifts from depressed to suicidal **Limitations:** • Limited to the university participants • Approach was only tested on SMS data

According to Table 2.1, all of the publications with the exception of works by Ji et al (2018), focused their research on a single social media platform only, either Twitter, Reddit, Weibo or Facebook. Besides that, it is also worth noting that the majority of the publications focused their research on binary classification only, in which tweets are classified as either suicidal or non-suicidal. Furthermore, while these publications share the same problem domain of detecting suicide ideation, only a minority of the works extended the scope to incorporate an intervention response upon the detection of suicide ideation. Lastly, it was observed that these publications used a balanced combination of artificial intelligence/ machine learning algorithms, NLP tools, feature extraction techniques to prove/ enhance the efficiency of suicide detection from social media posts, which will be explored in the following sub-sections.

2.2 Data Sources and Collection Methods

This section describes the method of data collection that are used across similar research works. The choice of dataset is a crucial step in the machine learning process, as the degree of data quality is one of the key elements in ensuring the accuracy of the prediction. The data source varies across different researchers, with some using public resources such as Twitter API and others focused on online forum postings that are written by people that have either

attempted or committed suicide. As such, data collection from sources such as Twitter, Reddit and Facebook will be explored in this subsection.

2.2.1 Twitter

Twitter is a real-time social networking website that allows their users to create and share posts (also known as Tweets) under 280 characters. When compared to other social media platforms, Twitter stands out as the best source of textual data due its ease of access and volume of data that can be collected. By default, a Twitter user's activity is visible to the public, regardless of whether the viewer is a Twitter user themselves. Twitter aims to provide a platform for users with similar interests and ideas to connect with each other, by sharing their views and using hashtags which usually contains the keywords of a topic of interest. Twitter's API allows the user to perform real-time monitoring on incoming tweets, accompanied with information that are associated with the Twitter user's profile such as: name likes and retweets.

The tweets are collected using a keyword filtering technique where the user specifies the words and phrases that are associated to the tweets they wish to collect. Besides that, geolocation data such as the latitude and longitude measurements that is captured when the tweet is posted can be accessed from the data collected through Twitter API. However, previous research works showed that only 1% of the tweets collected from Twitter's API contains geolocation information (Schlosser, Toninelli and Cameletti, 2021). In fact, the number of tweets associated with geolocation data have significantly decreased over the years (Tasse, et al., 2017). Furthermore, there might be inaccuracies with the geolocation information as the user might provide location information that are out of date or non-existent (Schlosser, Toninelli and Cameletti, 2021). Therefore, geolocation data might not be sufficient to contribute quality information to make smart inferences on the characteristics of the tweet.

Most of the research studies used Twitter's API to extract and build the dataset. For instance, Rabani, Khan and Khanday (2020) extracted the tweets based on a set of keywords and phrases that were used in similar research papers. Ji et al. (2018) created their own dataset by using Twitter API to extract tweets that contain words such as "die", "suicide" and "end my life".

However, further data pre-processing was required since some of the collected tweets were describing movies and advertisements which was thus irrelevant to suicide ideation.

2.2.2 Reddit

Reddit is an online forum that allows users of similar interest to interact with each other within community groups that are called subreddit. Users are free to share and view information related to their topic of interest within the subreddit. They interact with each other by voting and commenting on the different threads that are found in each subreddit. The main difference in the communication style between social networking platform and online forums is the anonymity of users in online forums. Unlike Twitter, Reddit users maintain anonymity with their user accounts as their actual private information such as name and age are not shared when they interact within the platform. Furthermore, Twitter users are more likely to connect friends from the real world (Ji et al., 2018). Therefore, users often maintain one time use accounts to improve anonymity as they do not want their accounts to be traced back to their real identity.

Data collected from Reddit is also widely used among research works as researchers leverage forum groups to perform sentiment analysis on specific domain areas. The subreddit “SuicideWatch” is popular among research works as the authors collect posts with suicide ideation through this Reddit thread. When compared to Twitter, most authors used datasets that were previously built by other researchers, instead of forming their own dataset from scratch like in Twitter. For instance, Ji et al. (2018) built a comprehensive Reddit dataset which consists of suicidal Reddit posts taken from “SuicideWatch” and non-suicidal posts taken from Reddit forums unrelated to suicide/ mental health for the automatic recognition of suicide intent through supervised learning machine model. In works by Muhammad Shah et al. (2020), the same dataset built by Ji et al. (2018) was used in their research to create unique feature sets that are passed to the machine learning model. (Tadesse et al., 2019) also utilised the same dataset to explore the detection of suicidal thoughts through hybrid deep learning models.

2.2.3 Facebook

Facebook is one of the major social networking platforms with an estimated of 1.13 billion active users per day. Compared to Twitter, content on Facebook has more privacy as the content is usually shared only to the user's friends and family. Therefore, this may encourage the user to be more open and honest in sharing content on the platform, thus suggesting that the usage of Facebook data may be more suitable in studying the mental state of the user (Calvo et al., 2017). However, Facebook data is notoriously known to not be research-friendly since it is tightly controlled under Facebook's privacy protocols. An information consent through research partnership with Facebook must first be granted in order to access specific Facebook data. The entire process from applying and approving the consent is a time-consuming process that is enough to make one lose hope in obtaining the data in the foreseeable future. In addition to that, Facebook impose specific undisclosed thresholds to restrict the data that is released through its public dataset, which indirectly limits the evaluation works of the research. This is as the inadequate data used in the research introduce bias in drawing conclusions. Thus, these complications have discouraged researchers to use Facebook data in their research works. In fact, at this point of writing, there were no research papers that are published in recent years that were found to be related to suicide ideation detection using machine learning or NLP techniques.

2.2.4 Summary

In comparison, the use of data that is obtained from Reddit and Twitter are most popular among research works due to the ease of access and availability of data. This gives research works an opportunity to leverage on the data to explore and test the feasibility of various analytical approaches. In addition to the main content, user data such as the interaction activity of the content and public user information such as the username and geolocation can also be obtained from Reddit and Twitter's public API. Therefore, this project integrated Twitter API to build the dataset that was passed into the machine learning model. The tweets were collected using the keyword filtering technique that is based on a set of keywords that are suicide-indicative. The tweets were also filtered and collected based on users that are located in the

United States and in English language only due to higher volume of tweets that can be collected from these users.

2.3 Natural Language Processing Method and Tools

Natural Language Processing (NLP) is a method of processing and retrieving information from natural language using a variety of techniques and tools in order to pass it as an input into a machine learning model. This is a crucial step as textual data is a form of unstructured data that is inconsistent in terms of format such as the length and language used. Thus, NLP will analyse and process the unstructured data in order to ensure that it is able to extract useful information during the feature extraction phase. The following is a list of common pre-processing tasks that are completed during sentiment analysis:

Table 2.2: Natural Language Processing Tasks

No	Task	Description
1	Chunking	Also known as shallow parsing, this feature identifies the grammar groups that the words belong to, such as verb, preposition and noun groups (Meyer, 2021).
2	Sentence Detection	Also known as sentence boundary identification, this function aims to detect the start and end of a sentence (Meyer, 2021).
3	Tokenization	Breaks down the sentence into smaller parts, which are also called tokens (Tadesse et al., 2019). The resulting tokens comprises of numbers, words or punctuation marks. The tokens can be normalized by removing numbers and punctuation marks and converting the words into lower case.
4	Name Entity Recognition (NER)	Identifies the named entities in the sentence based on pre-defined model categories such as names, locations, organizations, date and time. The outcome will indicate the tokens in the sentence which match the predefined class categories (Meyer, 2021).

Table 2.2 (Continued)

5	Lemmatization	Maps the word form to its basic form, commonly known as its 'Lemma' (Meyer, 2021). The lemmatizer will take the token and its grammatical tag as an input and output its lemma.
6	Language Detection	Determines the natural language of the given text (Meyer, 2021).

These pre-processing steps will provide structure, parse, tokenize and extract quality information from the textual data that is unstructured and arbitrary in nature.

2.3.1 Feature Extraction

Feature extraction is an important approach in machine learning since it extracts crucial information from the text input into a feature set that will be passed into the classifier (Ahuja et al., 2019). Thus, feature extraction has a direct influence on the model's performance (Wankhade, Rao and Kulkarni, 2022). Textual data consists of many different words that are considered as features to the machine learning. However, passing in the entire dataset without performing feature extraction results in a high volume of features that will create high dimensionality since some of the words could be redundant and contribute minimal support information. Therefore, feature extraction is crucial in order to find the important differentiating attributes of the data and transforms it into a feature set that will be passed into the machine learning classifier. The key is identifying the features that are able to provide high quality information and thus resulting in more accurate suicide detection. Hence, the 3 feature extraction techniques that will be explored in the following section are: Term Frequency-Inverse Document Frequency (TF-IDF), Bag of Words and POS tagging.

2.3.1.1 Term Frequency-Inverse Document Frequency (TF-IDF)

TF-IDF is a popular feature extraction method that is used to identify and evaluate the importance of a given term found in a document (Ahuja et al., 2019). The term frequency computes the number of times the given term

occurs in the document compared to the total number of words in the entire document. Then, Inverse Document Frequency (IDF) is applied to calculate the importance of the term. The formula of TF-IDF is broken down into 3 main steps. The complete formula is as follows:

$$TF = \frac{Number\ of\ times\ t\ is\ found\ in\ the\ document}{Total\ number\ of\ terms\ in\ the\ document} \quad (1.1)$$

where

TF = Term Frequency

t = Term *t*

$$\mathrm{IDF} = \frac{Total\ number\ of\ document}{Total\ number\ of\ t\ in\ the\ document} \quad (1.2)$$

where

t = Term *t*

IDF = Inverse Document Frequency

$$TF - IDF = TF + IDF \quad (1.3)$$

where

TF = Term Frequency

IDF = Inverse Document Frequency

TF-IDF is widely used as a feature extraction technique in research works as it can highlight unique words that represent important information since its given IDF is higher (Waykole and Thakare, 2018). Conversely, terms that appears frequently throughout the text but do not contribute any semantic importance can also be detected since its IDF is lower. For example, terms such as “as”, “is” and “are” have a low TF-IDF score to indicate its decrease in significance in representing important information. However, the limitation of TF-IDF is that it does not consider the syntactic meaning behind the term (Eklund, 2018). Therefore, synonymous words might be overlooked since it is categorised as 2 different words (Eklund, 2018). For instance, “unhappy” and

“upset” are counted as separate words although it is similar in terms of context. Contrariwise, homonyms such as “lie” which either refers to laying down or giving a false statement are counted as a one single term in TF-IDF.

In the works of O’Dea et al. (2015) found that using TF-IDF allowed its machine learning model to perform better when compared to other feature extraction techniques such as simple frequency. Similarly, Nobles et al. (2018) also proved that its machine learning model performed better in terms of accuracy when using TF-IDF as its feature set to measure the term frequency in each text message. This is attributed by the fact that the adoption of TF-IDF significantly reduced its incorrectly classified text message by an average of 28% (Nobles et al., 2018).

2.3.1.2 Bag of Words (BoW)

BoW is one of the easiest feature extraction methods as it computes the occurrence of each word in a given document (Wankhade, Rao and Kulkarni, 2022). The word and its associated word count will then be used to build a feature vector (Tadesse et al., 2019). The limitation of this approach is that the actual order of occurrence of the words will be lost since the vector of tokens is created in a randomised order. This can be solved by breaking down the words through bigrams (two words) instead of unigrams (single words), in order to retain the information stored in the natural order of the words. However, the resulting table will be very large, which ultimately results in highly complex computational efforts. Furthermore, similar to TF-IDF, the syntactic meaning of the text is not considered in this approach as it simply focuses on representing the sentence in vector form (Wankhade, Rao and Kulkarni, 2022). This will result in the dominance of words that frequently appear in the given text, resulting in domain specific words that hold greater significance but are less frequently used to be overlooked (Maken, Gupta and Gupta, 2019). Therefore, in most cases, TF-IDF are generally preferred over BoW due to its high performance (Wankhade, Rao and Kulkarni, 2022).

In the works of Rabani, Khan and Khanday (2020), applied both TFIDF and BoW in its feature extraction process to extract the most significant attributes in the text. Furthermore, they applied TF-IDF along with WEKA’s information gain algorithm to identify and remove terms of low

significance before the data is passed for training by the machine learning model (Tadesse et al., 2019).

2.3.1.3 Part-of-Speech (PoS) Tagging

PoS identifies and assigns the type of word that corresponds to each token. Each token is labelled with a grammatical tag such as noun, verb and adjective. This helps in reducing the ambiguity of the word, by normalizing it and giving us an understanding on the syntactic features that are associated with the given document. From there, the number of each grammatical tag that the word belongs to will be computed. For example, “I am depressed” may be tagged as: I: pronoun; am: verb; depressed: adjective. PoS tagging helps in sentiment analysis as an adjective is used more frequently to express the sentiment of the text (Wankhade, Rao and Kulkarni, 2022). In the works of Ji et al. (2018) used PoS as one of its feature sets that is passed into their suicide detection model.

2.3.1.4 Summary

Based on the review on the 3 feature extraction techniques, it is observed that each tool has its own strengths and limitations. Most authors use a combination of 2 or more of the above-mentioned tools to extract the important features from their dataset. Studies by Muhammad Shah et al. (2020) found that combining feature selection techniques along with best feature extraction algorithm helped to reduce the complexity and computational cost in addition to improving the machine learning model’s accuracy. Similarly, Tadesse et al. (2019) proved that applying a combination of handcrafted features such as Statistics, TF-IDF and BoW in their classification model achieved a higher accuracy when compared to including a single technique only. Ji et al. (2018) applied a combination of feature extraction techniques using tools such as LIWC, TF-IDF and PoS to obtain the statistical, syntactic, linguistic, word frequency, word embedding and topic features of the topic header and body text from the Reddit dataset that was used. Also using data obtained from Reddit, Shing et al. (2018) applied a combination of BoW to transform the data into vector form and TF-IDF to assign weights to the words. Therefore, the proposed feature extraction approach will use a combination of TF-IDF and PoS tagging to extract the features from the Twitter dataset. This

is as TF-IDF is more comprehensive when compared to BoW as it provides the score of the importance of each term, whereas POS tagging will be useful in extracting the syntactic features of the text.

2.3.2 NLP Tools

Sentiment analysis, which is a subcomponent of NLP, is an activity which aims to uncover the emotional perception behind a given text. There are 3 main approaches in sentiment analysis, which are lexicon-based, machine learning and hybrid approach (Hassonah et al., 2020). Lexicon-based approach counts the number of positive and negative terms that are found in a given text. From there, the sentiment of the text will be determined based on the majority of the terms that was found, for example: if the text had a higher number of positive terms, then it is concluded to be a positive sentiment. On the other hand, the machine learning approach involves the development of a machine learning classification model that is pre-trained with a dataset that is annotated with positive, negative and neutral sentiment. Then, the model is used to make predictions on the class label for an unknown class instance. For the hybrid approach, lexicon-based technique and machine learning are combined.

Majority of the research papers that were included in the literature review opted for the lexicon-based approach, by utilising NLP tools such as: OpenNLP and Linguistic Inquiry and Word Count (LIWC). On the other hand, Python's Natural Language Toolkit (NLTK) implements machine learning approach to perform sentiment analysis. Therefore, these 3 tools will be explored in the following sub section.

2.3.2.1 OpenNLP

OpenNLP is JAVA machine learning toolkit that is used for natural language processing, which aims to understand and extracts the sentiment behind the text content. OpenNLP offers 3 types of tokenization techniques: TokenizerME, WhitespaceTokenizer and SimpleTokenizer (Surve, 2019). TokenizerME requires a model file to be loaded in order to perform tokenization on the string. On the other hand, WhitespaceTokenizer and SimpleTokenizer does not require any model. The only difference between these 2 techniques is that WhitespaceTokenizer separates the tokens according

to the whitespace between it, while SimpleTokenizer splits the sentence into numbers, words and punctuation marks. OpenNLP offers 2 types of lemmatization techniques: statistical and dictionary based. The statistical method requires training data to build the lemmatizer model, whereas the dictionary-based method requires a dictionary file which consists of the word, PoS tag and its lemma.

In the work of Mbarek et al. (2019) used OpenNLP to perform sentiment analysis based on the Twitter profiles. They utilised the OpenNLP to analyse the number of positive and negative sentiment and terms that were associated with the Twitter user profile. Sentiment analysis was performed by comparing the content with the custom dictionary that contains positive and negative English terms that were collected by the authors from different sources.

2.3.2.2 Natural Language Toolkit (NLTK)

NLTK is an open-source Python library that offers independent modules that perform specific NLP tasks (Korani and Mouhoub, 2022). NLTK offers 2 types of tokenization techniques, which are sentence tokenization and word tokenization. As the name suggests, sentence tokenization splits the input data that consists of multiple paragraphs into sentences, whereas word tokenization separates the text data into individual words. In addition to the common NLP tasks, NLTK provides various easy to use features that further pre-processes the data. For example, NLTK provides an extensive library of stop words such as “was”, “that”, “is”, which are essentially words that are considered repetitive and does not contribute towards any useful information. Therefore, these words can be removed from the input data to reduce noisy data.

In addition to lemmatization, NLTK also provides stemming function which normalizes a given word to its root word. NLTK was introduced in the work of Tadesse et al. (2019), who utilised it to pre-process the training dataset by removing duplicated sentences, URL addressed and redundant words, as well as applying tokenization to separate the words into individual tokens. However, lemmatization was better than stemming in ensuring that the normalization process was sharp and efficient. This is as stemming would often result in hanging words that loses its meaning since the endings are drop

during the normalization process. For example, the word “Happiness” would be transformed into “Happi” if stemming was used. In comparison, this can be avoided if lemmatization is used, as the word will be directly mapped to its lemma which is “Happy”.

For sentiment analysis, NLTK has built-in machine learning operations that are powerful in deriving insights and extracting useful information from the textual data. For instance, Valence Aware Dictionary and Sentiment Reasoner (VADER) is NLTK’s pretrained sentiment analysis model that maps the input data to the lexical features and its associated emotional intensity measures. VADER does not require any training data and uses a rule-based approach as its sentiment reasoning logic (Hutto and Gilbert, 2014). Given an input sentence, VADER will first identify the words in the sentence with the lexicon dictionary. If a certain word found in the lexicon frequently appears in the sentence, then the grading of that particular word will be increased. Finally, the sentence will be graded based on its percentage of positive, negative and neutral sentiment. VADER also takes into consideration the use of capitalization and punctuation marks when it determines the sentiment score. It is best suited with social media data since it is efficient in recognizing slang and abbreviations that are usually found in short social media posts (Hutto and Gilbert, 2014).

2.3.2.3 Linguistic Inquiry and Word Count (LIWC)

LIWC is a text analysis software tool that enables the user to extract the underlying features of the text that are associated with emotional and cognitive components (Chung and Pennebaker, 2012). When compared with the popular NLP tools such as OpenNLP and NLTK, LIWC does not come with any pre-processing functions, as the main aim of LIWC is to perform sentiment analysis. In order to achieve that, LIWC scans through the text data word by word by mapping it with its internal dictionary and counts the percentage that a given word falls into each psychological category. This would help the user uncover the psychological states and behaviour of the author behind a textual data, which would then help towards the understanding of difficult and complex topics such as suicide (Chung and Pennebaker, 2012).

In studies by Liu et al. (2019), the authors utilised LIWC during the development of their suicide detection and response system titled Proactive Suicide Prevention Online. They utilised the SCLIWC tool, which is an enhanced version of the basic tool with support for Simplified Chinese language, to analyse the content that was written by the social media user before and after the response system was introduced. Primary evidence was found that the user's content had significant changes in their language, with reduce in death-oriented terms, coupled with an increase in future-oriented words after the response system was introduced. Besides that, in work by Sadilek et al. (2013), LIWC was utilised to study the mood pattern of the Twitter users based on 3 emotional states, which are: positive, negative and neutral. They were able to successfully identify temporal mood patterns within the Twitter users and make predictions on the emotional state that the user will be in within the next ten days.

2.3.2.4 Summary

Based on the 3 NLP tools that were explored, it is observed that all NLP tools except for LIWC offers various unique features in addition to the NLP tasks listed in table 2.2. In a study by Pinto et al. (2016) that aimed to compare and benchmark the performance of NLP tools including NLTK and OpenNLP, they concluded that there was no specific tool that performed significantly better in sentiment analysis when compared to the others. However, they did find that standard toolkits performed better in formal writings, whereas specialised tools that are targeted for specific social media platforms performed better in social media text (Pinto et al., 2016).

For the purpose of this research, NLTK was used for the pre-processing and sentiment analysis of the textual data. This is as NLTK's VADER has proven to produce outstanding results when performing sentiment analysis in social media text (Hutto and Gilbert, 2014). Furthermore, VADER is computationally efficient and economical as it can produce high quality results without comprising its accuracy (Hutto and Gilbert, 2014). Besides that, it is able to perform well in a wide range of domains due to its utilization of general sentiment lexicon as well as grammatical rules and syntax (Hutto and Gilbert, 2014). The features extracted during sentiment analysis will be useful

in filtering data, such that only tweets that are associated with negative sentiment will be passed into the machine learning model for classification

2.4 Machine Learning Model

In recent years, a considerable amount of research works has integrated machine learning models as part of their suicide prevention framework in an attempt to predict suicide risks. The model used in research works by is compared and summarised in the table below:

Table 2.3: Machine Learning Models

Publication	Model Used	Key Findings	Best Results
(Rabani et.al., 2020)	Naïve Bayes, Decision trees, Multinomial Naïve Bayes, Logistic Regression (LR), Support Vector Machine (SVM), Bagging, Random Forest, AdaBoost, Voting and stacking	• Random forest algorithm had the best performance • High precision and recall value due to application of feature extraction techniques	Accuracy=98.5% Precision=98.7% Recall=98.2%
(Mbarek et al., 2019)	Bayes Net, Adaboost, Sequential Minimal Optimization, J48, Random Forest	• Random forest had the best performance • Adding more features like linguistic, emotional and facial into the model helped increase performance of the model	Precision=83% Recall=83% F-measure=83%

Table 2.3 (Continued)

(O'Dea et al., 2015)	SVM and LR	• SVM with TF-IDF and no filtering of stop words performed the best. • Model unable to achieve learning plateau when more data is added	**Strongly concerning** Precision=80% Recall=53% F-measure= 64% **Possibly concerning** Precision=76% Recall=91% F-measure= 83% **Safe to ignore** Precision=75% F-measure= 63% Recall=53%
(Liu et al., 2019)	SVM, Random Forest, Decision Tree and LR	• SVM had the best results	Precision=88% Recall=85% F-measure= 85%
(Tadesse et al., 2019)	Convolutional Neural Network and Long Short-Term Memory	• Combination of model outperform single classifiers • Combination of feature extraction techniques (statistics, TF-IDF, BOW) on single machine learning classifiers gave comparable results (Accuracy score: 82%)	Accuracy=93.8% Precision=93.2% Recall=94.1% F-measure= 93.4%

Based on the research papers included in the literature review, supervised learning was mostly used for the classification of the data instances, in which a set of pre-annotated training data is used to build the machine learning model for the classification task. It was also observed that the 3 most used machine learning algorithm were Random Forest, SVM and LR, where all 3 algorithms were included in 3 out of 5 research papers in Table 2.3. Furthermore, Random Forest and SVM performed significantly well and achieved more than 80% in all 3 of the performance metrics that were used. Hence, the implementation, strengths and limitations of Random Forest, SVM and Logistic Regression will be further explored in the following section.

2.4.1 Random Forest

Random Forest is a supervised machine learning algorithm that is commonly used to solve regression and classification problems. It is essentially an ensemble of decision tree algorithms that uses a divide and conquer approach to improve the performance of the model (Nordin et al., 2021). The algorithm will first create different bootstrap subsets of the training dataset. Then, the model will construct individual decision trees based on the training dataset. Each of the individual decision tree will then vote for a data point. The final decision of the model will be based on the result that has the majority voting from each individual decision tree (Boudreaux et al., 2021).

Table 2.4: Strengths and Limitations of Random Forest

Strengths	Limitations
• Able to perform well with large data sets that contains large number of attributes and missing data (Dineva and Atanasova, 2020) • Able to adopt dimensionality reduction method to filter and identify the most significant variables in the dataset, thus reducing the risk of overfitting the model	• Not suitable for regression problem, may risk overfitting if the data is too noisy • Longer time and a large amount of memory needed to train the data (Dineva and Atanasova, 2020)

2.4.2 Support Vector Machine (SVM)

SVM is also a supervised machine learning algorithm that aims to identify an optimal decision boundary that accurately differentiates the 2 classes of data points. In order to achieve that, SVM analyses the data and finds the hyperplane with a margin that has a widest distance between the dividing lines that separates the different classes of data (Nordin et al., 2021). An optimum hyperplane will improve the model's ability to effectively classify new data instances as the prediction of class label is determined by which side of the hyperplane that it falls on (Boudreaux et al., 2021).

Table 2.5: Strengths and Limitations of SVM

Strengths	Limitations
• Performs well on regression problems as the effect of SVM increases along with the dimensional space (Dineva and Atanasova, 2020)	• SVM is not able to perform efficiently with noisy data as it extensively uses cross-validation to improve its computational efficiency. Thus, leading to poor model performance (Dineva and Atanasova, 2020)

2.4.3 Logistic Regression (LR)

LR is a supervised machine learning model that uses a logistic function to predict the output of the dependent variable by using a set of independent variables (Nordin et al., 2021). The function will estimate the probabilistic value of the class the data is classified to.

Table 2.5: Strengths and Limitations of LR

Strengths	Limitations
• Easy to implement and train model with dataset with minimal number of attributes (Dineva and Atanasova, 2020)	• Unable to perform well on large datasets, due to its high dimensionality (Dineva and Atanasova, 2020)

2.4.4 Summary

In summary, each of the 3 machine learning algorithms: Random Forest, SVM and LR comes with a set of strength and limitations that address different types of data modelling challenges such as noisy data, large datasets and ambiguous data. Hence, the machine learning algorithm should be selected based on its ability to produce good performance results. Considering the nature of the training data that is noisy and has large number of attributes, Random Forest was as the algorithm during the machine learning phase, since it is proven to have performed well on similar datasets with such characteristics.

2.5 Existing Works

In recent years, there are various research works that seek to establish and strengthen the basis in identifying suicide ideation through machine learning and deep learning approaches. In this section, a total of 5 research works were reviewed in terms of the dataset, model and approach used. The research work reviewed in this section are listed in Table 2.6.

Table 2.6: Comparison of Existing Works

No.	Publication	Origin of Dataset/ Reviewed Platform)	Response System
1	Rabani, Khan and Khanday (2020)	Twitter	No
2	Ji et al. (2018)	Reddit and Twitter	No
3	Liu et al. (2019)	Weibo	Yes
4	Yang et al. (2021)	Weibo	Yes
5	Gomes de Andrade et al. (2018)	Facebook	Yes

Firstly, research works from Rabani, Khan and Khanday (2020) and Ji et al. (2018) that focuses on exploring the feasibility of suicide detection within social media platforms were reviewed. An extension to this research question includes the development of a response mechanism as a means to manage the individuals at risk that are detected by the algorithm. In connection with that, the research works of Liu et al. (2019) and Yang et al. (2021) were reviewed. Finally, the past responses mechanism developed by Samaritans and

current response mechanism used by Facebook is compared and evaluated in the final part of this section.

2.5.1 Rabani, Khan and Khanday (2020)

Rabani, Khan and Khanday (2020) explored the feasibility of identifying tweets that contains suicide ideation using machine learning and ensemble methods. The dataset was built by extracting tweets from Twitter API based on keywords that are related to suicide ideation. From there, the collected tweets were manually annotated by mental health researchers into 2 classes: 1: suicidal; 2: non-suicidal. After data pre-processing and feature extraction, the dataset had a total of 4266 tweets that were passed into different machine learning and ensemble methods for classification. This research work provided researchers an understanding of how social media data can be utilized to extract key information that indicates the user's emotional distress linked to suicidal ideation. However, since the authors utilised the WEKA tool, which is a GUI based machine learning software to validate its approach, there was a lack of description and details on the data pre-processing, feature extraction and data modelling techniques applied to achieve their objective.

2.5.2 Ji et al. (2018)

Ji et al. (2018) explored the use of supervised learning to detect suicidal ideation from social media posts on both Reddit and Twitter. As such, 2 separate datasets were built, which consisted of 3549 and 102888 samples respectively. This study built their features set based on an extensive collection of features which includes statistical, linguistic and syntactic features. A variety of classification models were built to compare its classification performance. This research work provided a basis that validates the application of feature processing and classification methods for suicide detection in Reddit and Twitter. Furthermore, this study provided an understanding on the linkage between the words, language and topics associated with both suicidal and non-suicidal textual posts. However, this understanding is limited to binary classification only. Moreover, this study did not explore the linkage between the feature sets and its effect to the performance obtained by the classification model.

2.5.3 Liu et al. (2019)

In research works that involve the development of a response mechanism, Liu et al. (2019) used data collected from the comments section of a post shared in Chinese social networking site, Weibo that was written by a user named "Zou Fan" who committed suicide in 2012. The authors explored the acceptability and feasibility in identifying individuals that are at risk of suicide through various machine learning models. If it is detected that the individual is at risk of suicide, a direct message will be sent to the user offering helpful support information and specialised crisis management that is attended by certified and experienced counsellors. This ensures that the support provided through this system is tailored to the individual's needs. It was concluded that the mechanism was effective as the number of death-oriented word mentioned in the user's microblog posts significantly reduced after the intervention of the counsellors. However, the approach was not comprehensive enough as only primary intervention response was given towards the at-risk individual which first requires their permission to opt-in and active participation in order to be effective. Furthermore, because the primary goal of this work was to report and validate the overall efficacy of implementing a suicide response mechanism, there is a lack of details on the data pre-processing, feature extraction and data modelling approach used in the research work.

2.5.4 Yang et al. (2021)

Yang et al. (2021) reviewed the development of the suicide screening and crisis response program, titled: 'Tree Hole Action" that integrates advanced Artificial Intelligence algorithms and mental health services for real-time monitoring of suicide risk. Using a mix of knowledge graph and Definitive Clause Grammars Transformation Rules, the comments section of the same Weibo post used in research work by Liu et al. (2019) was analysed in real-time using natural language processing techniques to extract data attributes for further analysis. The "Tree Hole Action" adopts a systematic approach for suicide screening as a report will be generated for the comments that are associated with a suicide risk beyond level 6 (based on a scale of 1-10) and sent to a team of trained volunteers to initiate a targeted crisis intervention response. Individuals with risk level below 8 will be offered counselling

services, while the volunteers will attempt to contact the family and friends of individuals beyond the risk level of 8 to share information regarding their suicidal behaviour. In addition to that, all high-risk individuals will be placed under a monitoring list where their account activities will be closely monitored and actively contacted by volunteers to assess their suicide ideation through continuous communication and supervision. A rescue team is formed to mobilise efforts to detect the location of the target, in the event that a strong suicide ideation is found from the high-risk individual.

Through this research work, it was proven that the combined use of AI and standardized crisis management to provide practical, timely and effective support marks a new turning point in the development of suicide monitoring and response strategies. Among the critical success factors of the strategy reviewed by the author includes the integration of online and offline support by the volunteering team ensures that the targeted individual receives active and timely support through continuous communication and supervision by the rescue team. The success rate from this approach indicates its feasibility and sustainability in the long run towards better suicide prevention management.

2.5.5 Gomes de Andrade et al. (2018)

Facebook attracts billions of users each day on their platform, with users that choose to express their mental distress and suicidal thoughts. This puts Facebook in a critical position to help connect distressed individuals to their friends and families who can support them (Gomes de Andrade et al., 2018). In recognition and with full agreement to this statement, Facebook has partnered with various mental health organizations like National Suicide Prevention Lifeline to understand the critical aspects and granular details that are crucial towards building a suicide prevention mechanism that can better manage the social and personal issues that their users may be going through. Addressing these calls, Facebook's very own AI powered suicide prevention tool was rolled out to the platform in 2017. It takes the contents and comments under the post as an input in their Random Forest classification model to determine if the post under review contains any suicidal ideation. Posts that require urgent review will be passed through Facebook's internal Community

Operations team for their evaluation on whether the local authorities should be alerted.

Although the suicide detection tool has faced backlash by experts due to data and privacy concerns, Facebook addressed these critiques with justification that their motivations were backed up by extensive research and support by various suicide prevention organizations to formulate a standardized process that will be implemented across Facebook (Gomes de Andrade et al., 2018). Furthermore, the suicide prevention tool is also aligned with their corporate ethics and mission to manage the safety and well-being of the communities within Facebook. Due to the elevated privacy expectations associated with these contents, Facebook has decided to exclude content shared on secret groups and posts with privacy settings set to "Only Me" from being analysed by their suicide detection algorithm in order to balance the privacy and efficacy concerns of their suicide prevention tool. In addition to that, Facebook also limits their response to be seen by the at-risk individual only. Unlike Samaritan's Radar that alerted the friends of the individuals at risk, Facebook seeks to only target author of the post with recommendation on the mental health resources if suicide ideation is detected by their machine learning classifier.

On top of that, social media sites such as Facebook have collaborated with charity organizations such as Samaritans to initiate a suicide alert reporting system for users to report other users who they find to express suicidal behaviour through their online content. The main intent of this self-reporting tool is to minimize the gap between detecting posts with suicidal intent and providing the necessary help the user at risk needs. User reported posts will be taken into consideration with higher priority by the moderator when it is passed for review. However, the reality is that online users are either unlikely to act on a suicide note once they come across it (Soron and Shariful Islam, 2020). Furthermore, there are also situations where the friends and followers ignore the red flags and instead further provoke and encourage the person at risk to commit suicide by leaving harsh comments and negative reactions to the post.

2.5.6 Summary

Based on the research works by experts throughout the years, it is observed that most studies are only limited to binary classification only. Furthermore, while there are many published studies that aims to detect suicide intent through online social media through various machine learning approach, there is a significantly lower number of studies that looks into the development of a complete suicide detection and response framework that actively monitors social media posts in real-time. Moreover, the research works reviewed within this section have a limited scope in which their studies explored either the detection or response of suicide ideation only, but not both. This shows that there is an indeed lack of research undertaken to develop a comprehensive and proactive mechanism towards the detection of different degrees of suicidality and tailored intervention upon detection of such ideation.

Therefore, this provides an opportunity for the development of the proposed framework, which aims to analyse and detect suicide intent through textual posts found on Twitter through a multi-class machine learning classification model in real-time. The proposed solution is targeted towards users on Twitter platform, as an extension to enhance the current suicide response system that is in place. After classifying the level of suicide risk associated with each tweet, a direct message with informational mental health resources is delivered to individuals who have tweets with medium to high risk levels. Further to that, posts with high level of suicide risk will also be flagged for further review by an internal moderator that ideally has general or specialised knowledge on mental health. The moderator will next evaluate if immediate action is required to alert the local authorities.

As described previously, there are various privacy concerns regarding the adoption of an automated suicide prevention tool within social media applications, which inevitably prompted debates on the ethical nature of the tool. Despite that, efforts by Facebook and the “Tree Hole Action” proves the efficacy and sustainability of the tool in offering social support and addressing the issue of suicidal behaviour on social media application. As there are no definitive answers or correct solutions that completely solves ethical issues, the best way forward is to strive for a balance between the interest to address this issue and long-term values associated with the mission towards the

sustainability of the suicide prevention tool. With that consideration in mind, the guiding principle of the tool should be established based on the aim to contribute towards protecting the well-being and safety of the online community as a whole. Leveraging the ethical motivation that drives “Tree Hole Action” and Facebook’s approach towards their success in their suicide screening and response framework, the proposed approach aims to provide a fully transparent solution with a comprehensive underlying thought process to shed light and lower the barriers of individuals experiencing mental health crisis to seek professional support.

2.6 Software Engineering Methodologies

In a typical Software Development Lifecycle (SDLC), software project activities are categorised into 6 main phases which are: planning, analysis, design, implementation, testing and maintenance. This provides a well-structured and standardised approach in developing high quality software that meets the customer’s expectations. In order to further improve the quality of the software, various software engineering methodologies are implemented to provide a framework that further refines and tailors the sequence of the 6 main phases to better suit the project nature and deliver quality software in a timely, efficient and effective manner. Therefore, this section will explore 3 different software engineering methodologies which are Waterfall, Rapid Application Development (RAD) and Scrum approach.

2.6.1 Waterfall

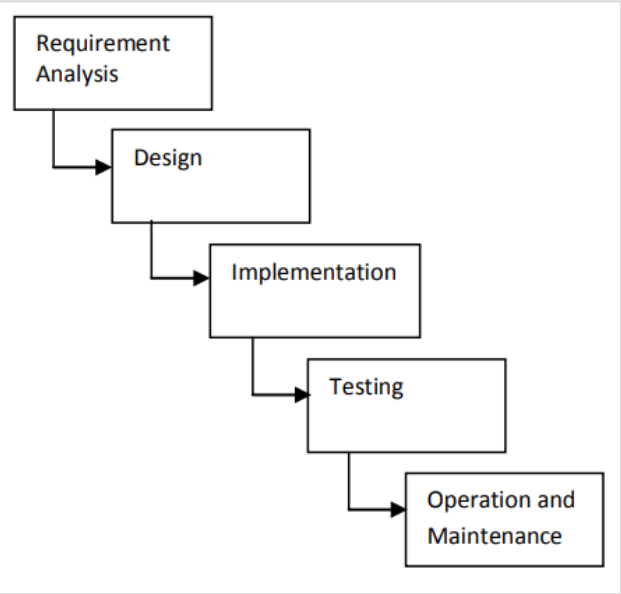


Figure 2.1: Waterfall Approach (Pfleeger and Atlee, 2006 citied in Adenowo and Adenowo, 2013, p.429)

Waterfall methodology uses a plan-driven approach which involves planning ahead and subsequently implementing the plan in a linear and sequential manner for the entire project phase (Adenowo and Adenowo, 2013). As illustrated in figure 2.1, in order to progress from one phase to the other, the deliverables by the end of each phase must first be accomplished, which typically requires undergoing the process of review, verification and approval by the client.

Waterfall helps to coordinate the workflow for large software projects that usually involves members from different teams, departments or area of expertise. This is mainly because waterfall methodology heavily emphasizes on the need to follow a well-defined and structured approach to accomplish each project milestones (Adenowo and Adenowo, 2013). This allows the team to follow a standardized and systematic workflow which boosts the coordination between different team members. Besides that, since the framework involves certain degree of path-dependence to complete one or more project deliverable, this helps to coordinate the work sequence and productivity of the team members (Adenowo and Adenowo, 2013). Therefore, waterfall methodology is best suited to be applied for projects which follows a

strict set of user requirements that are clearly defined and unlikely to undergo any changes throughout the software lifecycle.

2.6.2 Rapid Application Development (RAD)

RAD is a methodology that emphasizes on the production of high-quality information system. RAD is driven by 4 main concepts: iterative and quick prototyping, active user participation through constant feedback, developing reusable code and delivering the working software as soon as possible. In order to accelerate the software development process, changes to the system design or logics are rapidly incorporated into the prototype, without having the need to restart the entire software development lifecycle from scratch (Qudus Khan et al., 2020). Hence, the overall duration of the project is minimised since the time spent during planning and design phase is significantly reduced.

RAD is best suited for short term projects whereby the working model needs to be shown to the customer within a short period of time. This is mainly due to the framework's highly iterative nature which supports rapid development (Qudus Khan et al., 2020). RAD provides the flexibility for the prototype to be adjusted according to additional requirements that are discovered throughout each iteration of the prototype, rather than starting the project with a strict, fully known set of requirements. Therefore, midstream modifications are embraced as it can be incorporated seamlessly into the application. Furthermore, similar to scrum methodology, RAD breaks down the project into smaller and manageable tasks. Besides streamlining the activities involved in the project, this helps the project team members to work through the complicated parts of the project more easily as feedback can be obtained from the product owner to ensure that the working model or prototype meets the user's requirements and desired system process (Qudus Khan et al., 2020). Hence, this will improve team productivity as the project blockers are addressed and quickly resolved once it is identified.

2.6.3 Scrum

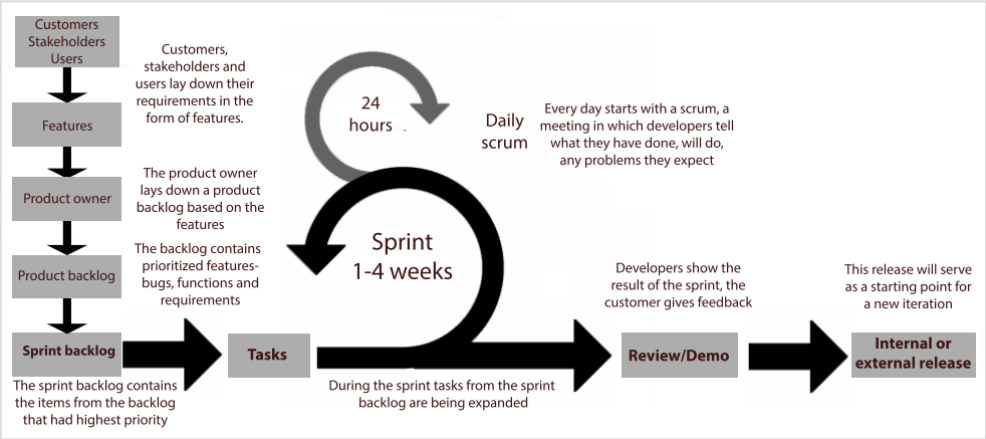


Figure 2.2: Overview on Scrum Methodology (Van Casteren, 2017)

Scrum methodology adopts agile principles by using an iterative and incremental framework whereby the features are worked on by sprints (Zikopi, 2019). During each sprint, the team members that are led by a scrum master will maintain a close collaboration with each other to deliver the work product by the end of each sprint. Scrum stands out as one of the most widely used agile framework since it strives to keep the team focused on making progress incrementally throughout each sprint. This is achieved through the adoption of agile rituals such as sprint planning, daily scrum, sprint review and sprint retrospective (Zikopi, 2019). These activities provide a platform for the project team members to keep the team engaged with each other's progress in an effective manner. As such, scrum helps to elevate the efficiency of the software development process in the aspects of timeliness and quality of the software delivery.

Scrum is best suited for software projects that develop complex products with the goal of delivering business values incrementally over a short period of time (Zikopi, 2019). This is as the scrum framework breaks down the development of modules into short and maintainable sprints that collectively constitutes the final, complete software product. This also ensures that the activities defined during each sprint can be completed within a specific timebox. Furthermore, scrum framework emphasizes on self-management by encouraging continuous self-improvement and workflow transparency within the team members (Zikopi, 2019). Through daily scrum meetings, the team

members are able to gain a clear perspective on their tasks and work more efficiently to produce quality deliverables.

2.6.4 Summary

Table 2.7: Comparison of Software Development Methodologies

Software Development Methodology/ Qualities	Waterfall	RAD	Scrum
Main Objective	Focused on following a linear and sequential approach to achieve project milestones	Focused on developing working prototypes quickly for client feedback	Focused on developing and delivering features in time-boxed sprints
Ideal Project Duration	Medium-Large; large project team that consists of several teams that handles different aspects of the project	Small-Medium	Small, Medium or Large; small multidisciplinary teams that are equipped with domain knowledge across many areas
Characteristics	• Emphasizes on accomplishing each project milestone in a formal, systematic and sequential approach	• High level of customer involvement during the entire SDLC • Goal is to meet the business needs of the system by rapidly incorporating changes to the software	• Emphasizes on self-discipline and continuous improvement of team members • Goal is on efficiency by working on different sets of features within a short period of time

Table 2.7 (Continued)

Strengths	• Structured and fully documented process • Easy management-defined goals and review process	• Delivery time is fast • Client satisfaction • Rapid delivery of prototype • Adaptive and flexibility in implementing changes	• Increased productivity, allow sprints to run in parallel • Lower risk as issues are identified at earlier stage

Based on Table 2.7, it is observed that although both Scrum and RAD are similar in terms of its iterative nature, the main difference is that scrum places more consideration on completing the design of the product, while RAD is more focused on the capturing the complete functionality of the prototype. Therefore, scrum methodology was used as the development methodology of the proposed approach. Scrum's emphasis on completing features in an incremental manner is aligned with the project's intent to deliver the final complete system in a progressive manner. Furthermore, the organization of the work activities into time-boxed sprints helps coordinate the workflow and ensure that the final system is completed within the limited project duration, without compromising its quality.

2.7 Web Application Framework

2.7.1 React

React is a JavaScript library that is used for building dynamic and interactive User Interface (UI) for web applications in an efficient and structured manner (Maratkar and Adkar, 2021). In other words, React handles the front-end design of the web application in terms of its structure design and appearance. The UI development is component based, whereby each component is responsible for rendering an independent piece of content. The component collectively forms the building blocks of the web application, as it is reusable and can be nested within each other to form a complex application. The primary advantage of using React is its ability to increase performance by optimizing the load time of the application (Maratkar and Adkar, 2021).

Furthermore, the modularisation of the components promotes code reusability, which would then reduce the complexity of the code and development effort.

2.7.2 Laravel

Laravel is an open-sourced web application framework that provides expressive and easy to use syntax to support the development of robust web application development that follows the Model-View-Controller design concept (Bagwan and Ghule, 2019). Laravel provides a variety of features such as application logic, RESTful controllers and blade templating that can be used to facilitate the development of the application. Furthermore, Laravel provides support for external client-side scripting frameworks such as React.js, Bootstrap and Vue.js to develop the UI of the web application.

2.7.3 Flask

Flask is a Python-based web framework that is commonly used as a third-party library in the development of web applications. Flask is considered as a microframework since it does not provide extensive features such as the abstraction of database through Object Relational Mapping. Its main aim is to provide support for the development of simple and scalable applications. Due to the nature of the framework that is a versatile and easy to use and open-sourced, it is best suited for small-scale and real-time applications. This is mainly attributed to the fact that Flask utilises Jinja2 Template Engine for the development of dynamic html web pages. Similar to Laravel, Flask provides a variety of functions that ease the creation of RESTful APIs. Hence, this Flask is good for the development of powerful and high performing web application.

2.7.4 Summary

Based on the 3 web application frameworks explored in the section above, Flask was used in the proposed approach due to its ability to support HTTP requests and easy integration with Python libraries. This will allow the integration of the machine learning model as well as fetching the tweets in real-time through Twitter API. Furthermore, since the proposed approach intends to build a one-page web application only, the simple UI and dynamic

functions by Jinja2 Template Engine will be sufficient to support the development of the web application.

CHAPTER 3

METHODOLOGY AND WORK PLAN

3.1 Introduction

Scrum methodology was implemented in this project, given the nature of the project where adaptability and flexibility to implement change is crucial in order for it to succeed. The project duration runs for a total of 28 weeks, whereby the first 14 weeks took place from January to April 2022, while the remaining weeks were a continuation of the project that runs from June to September of the same year. There were a total of 3 sprints in this project, with each sprint set to run for a total of 20 days excluding weekends. Scrum meetings was held on every Thursday as a progress checkpoint to synchronise the activities that have been performed, are being completed and will be completed. These scrum meetings serve as a feedback loop to adapt and optimize the workflow according to the current progress. Apart from that, sprint review and retrospective were carried out to solicit feedback on the working product delivered during the end of each sprint. This would also provide an opportunity to periodically review and reprioritise the deliverables that are planned for the next sprint.

3.2 Software Development Methodology

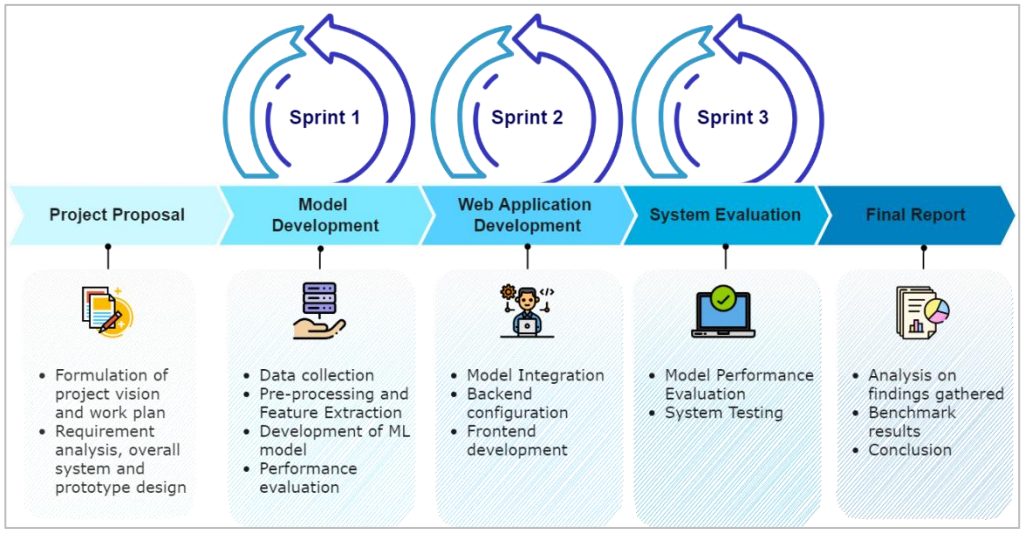


Figure 3.1: Overview on Scrum Methodology used

Figure 3.1 depicts an overview of scrum methodology that was applied in this project. In the first phase, the project proposal was prepared, which covers all activities involved in the initial planning and design of the project. This includes the analysis and formulation of the project vision and project scope, literature review as well as the design of system architecture, use case and prototype. Upon completion of the initial proposal phase, there are 3 main sprints that covers different areas of development in this project. The first sprint includes all activities involved in model development, such as data collection and pre-processing as well as the model training and testing. The following sprint focuses on the development of the web application, which involves activities such as the configuration of the backend, model integration and development of the frontend UI. From there, the third and final sprint focuses on the evaluation of the entire working system in which the performance metrics was measured and benchmarked to evaluate the overall success of the project. Once completed, the project entered the closed off stage in which the findings gathered were documented in the final project report.

3.2.1 Initiation

In the project proposal phase, a preliminary study on the project was first conducted in order to establish the project's objectives. Prior to that, the problem's background was investigated, in which the root cause of the problem was investigated, providing context for determining the problem statement. Based on the problem statement, the project objectives were then outlined to provide a clear direction for the project and align the activities needed to achieve its final outcome. From there, the proposed solution, proposed approach as well as the project scope were defined for this project. During which, an exploratory analysis was carried out to identify and compare the different types of approach, development tools and software development methodologies used across similar research works. This is to identify the necessary components and provide overview on the solution required to deliver the overall system that meets the project objectives. Thereafter, a system architecture diagram was created to provide a high-level overview of the entire system by mapping out all the components that will be delivered

throughout the course of this project. The system architecture diagram helps to define the project scope as it provides the necessary information to determine what type of features will be and will not be included within the deliverables of the project.

3.2.2 Planning

Aside from the major deliverables of the project, the project scope also defined the assumptions and constraints of the project. Once completed, the work plan was created which consists of the Work Breakdown Structure (WBS) and Gantt chart of the project. This serves as guideline to monitor the project activities and ensure that all planned deliverables were completed within the given timeline. Once the project vision was established, the requirement analysis was carried out. Academic works in relation to the project's objectives were reviewed, analysed and evaluated to identify the trends, challenges and gaps of the topic area. A total of 12 research papers were selected for review based on their data collection method, NLP techniques and tools, machine learning model, existing system, software development methodology and web application framework used. Essentially, these findings contribute towards strengthening the basis of the proposed solution and approach of the project. From there, the vision was refined in order to incorporate the findings and ideas derived from the literature review to create a comprehensive system.

3.2.3 Design

The detailed design of the system's architecture, which specifies the model, system requirements, use case and development tools was established during the design stage. The model of the system was mapped out to provide a detailed description behind the classification logics used in the machine learning model. This gives an overview on how the model classifies the data instances into different levels of suicide risk and provides the prediction as its output. In addition to that, details on performance evaluation metrics were also defined in order to ensure that the model is able to achieve satisfactory performance result. Then, the functional and non-functional requirements as well as the use case were defined and documented to establish the system's

requirements that will be delivered during the development sprints. Once completed, a preliminary design was created to show how the requirements are captured in the conceptual design of the machine learning model and web application prototype.

3.2.4 Development

The development stage is broken down into 3 sprints: model development, web application development and system evaluation. In the first sprint, the development of the machine learning model was carried out. It begins with the formation of the dataset, which consists of tweets that will be passed into the machine learning model. Due to the limitations of Twitter API, which only allows the lookup of 900 tweets every 15 minutes, the data collection period was carried out for a total period of 1 week to maximise the number of tweets that can be collected within this period. Each lookup request captures the tweet content, username, geolocation, time and date when the tweet is posted. Once the dataset was built, each tweet was first manually annotated according to its level of suicide risk. Then, data pre-processing was performed, which includes a variety of data cleaning tasks to reduce the noise of the dataset. Natural language processing was also performed to build the feature set which consists of syntactic, linguistic and sentiment features. Then, the machine learning model was created. The feature set is then passed into the model for training and testing.

In the second sprint, the implementation of the web application development was carried out. Firstly, the backend of the application is configured using the Python library, pickle to import and integrate the trained machine learning model with the web application. Then, the setup of the API endpoints was defined and configured to allow the input from the web application to be passed into the model, as well as the model prediction to be passed back to the web application. Finally, the user interface of the web application was developed, which provides a simple and lightweight design that comprehensively captures the use case requirements that were specified during the design phase.

3.2.5 Testing

Once the development and setup of the web application was completed, the project enters the third and final sprint, whereby system's performance was tested and evaluated. Evaluation metrics was used to assess system's ability to accurately classify the level of suicide risk based on the tweets. The performance results obtained during the model testing phase were analysed and benchmarked with existing works to determine if the proposed solution was able to meet its defined objectives of improving the classification performance. Finally, system test was conducted on random test samples to validate the efficacy of the system towards real-time data, as well as to check if the correct response was triggered for each suicide risk level.

3.2.6 Close-off

During the close-off stage, the final year project report was prepared, whereby the findings and conclusion of the entire project were documented. Once completed, the report was submitted, which marks the end of the project.

3.3 Development Tools

Several development tools were needed in the development of the complete proposed system. The 3 main development tools that were used includes Twitter API, Jupyter Notebook and Flask.

3.3.1 Twitter API

Twitter API enables the real-time monitoring and collection of incoming tweets, accompanied with information such as its tweet's date and time, author and geolocation. Twitter API offers a variety of functions that provides developers the flexibility to lookup the tweets based on the characteristics of tweet or keyword used in the tweet. In the proposed solution, Twitter was heavily utilised in 2 main aspects. Firstly, the initial dataset was collected through the initiation of lookup requests on the API, which were used to build the training and testing dataset of the machine learning model. Secondly, Twitter was utilised once again during system test, in which the test samples used are random tweets captured from Twitter API for further processing through the system.

3.3.2 Jupyter Notebook

Jupyter Notebook is an open-sourced, web-based Integrated Development Environment (IDE) that is widely used in the field of Data Science and Data Analytics since it supports the execution of code through line by line. This allows the users to present in-line visualization and run code snippets within their own browser. Furthermore, it supports programming languages such as Python, R and Scala. In this project, Jupyter Notebook was used as the IDE platform to develop and train the machine learning model prior to its integration with the web application. In order to train the machine learning model, the annotated dataset was loaded, cleaned and pre-processed using several Python libraries. The detailed listing of the Python libraries utilised in this project are listed in Table 3.1.

Table 3.1: Python Libraries Used

No	Library Name	Description
1	RE (Regular Expression)	• Used during data pre-processing to remove redundant words and characters
2	Pandas	• Used to access, analyse and manipulate the dataframes that were created to load and process the annotated dataset
3	NumPy	• Used to access and manipulate the temporary arrays that were created for data pre-processing and feature extraction
4	NLTK	• Mainly used during the data pre-processing and feature extraction phase to clean and transform the dataset into high quality feature sets • This includes the tasks for tokenization, lemmatizing, pos tagging and computing the sentiment score
5	Sklearn	• Mainly used for 3 different functions: i. To extract the TF-IDF score from the dataset using the *TfidfVectorizer* module ii. To build the Random Forest Classifier iii. To obtain the performance results using the *classification_report* module, as well as visualize the model's performance using *confusion_matrix* module
6	Seaborn and Matplotlib	• Used to build intuitive and vibrant statistical graphs to visualize and gain insights on the data

Table 3.1 (Continued)

7	Pickle	• Used to serialize the final, trained machine learning model from Jupyter Notebook to be deployed on the web application

3.3.3 Flask

Flask is the main framework used to develop the web application. On the backend, the scripts are developed to integrate the machine learning model from Jupyter Notebook, perform data pre-processing and feature extraction, as well as establish the API connection to process the real-time data for prediction. To achieve that, a Python virtual environment was set up to deploy the pre-trained model using the *pickle* library. Then, the pre-processing functions were defined to clean and build the feature set from the real-time tweet. From there, API endpoints were configured to facilitate the interaction between the real-time input captured from the web application and the prediction that was output from the model.

3.3.3.1 Frontend Template Engine and Libraries

On the frontend, Jinja2 was used as the main templating engine to build the dynamic web page that is rendered with the data passed to the template from the backend. Besides that, Bootstrap was used to build a simple and intuitive interface for the moderator to review and evaluate the high-risk tweets that were flagged by the machine learning model. Furthermore, logical functions were configured using Javascript to represent the delivery of the default direct message to the target at-risk user. Lastly, Chart.js was used to build the visualization charts that displays the real-time statistics of the analysed tweets.

3.4 Work Plan

3.4.1 Work Breakdown Structure (WBS)

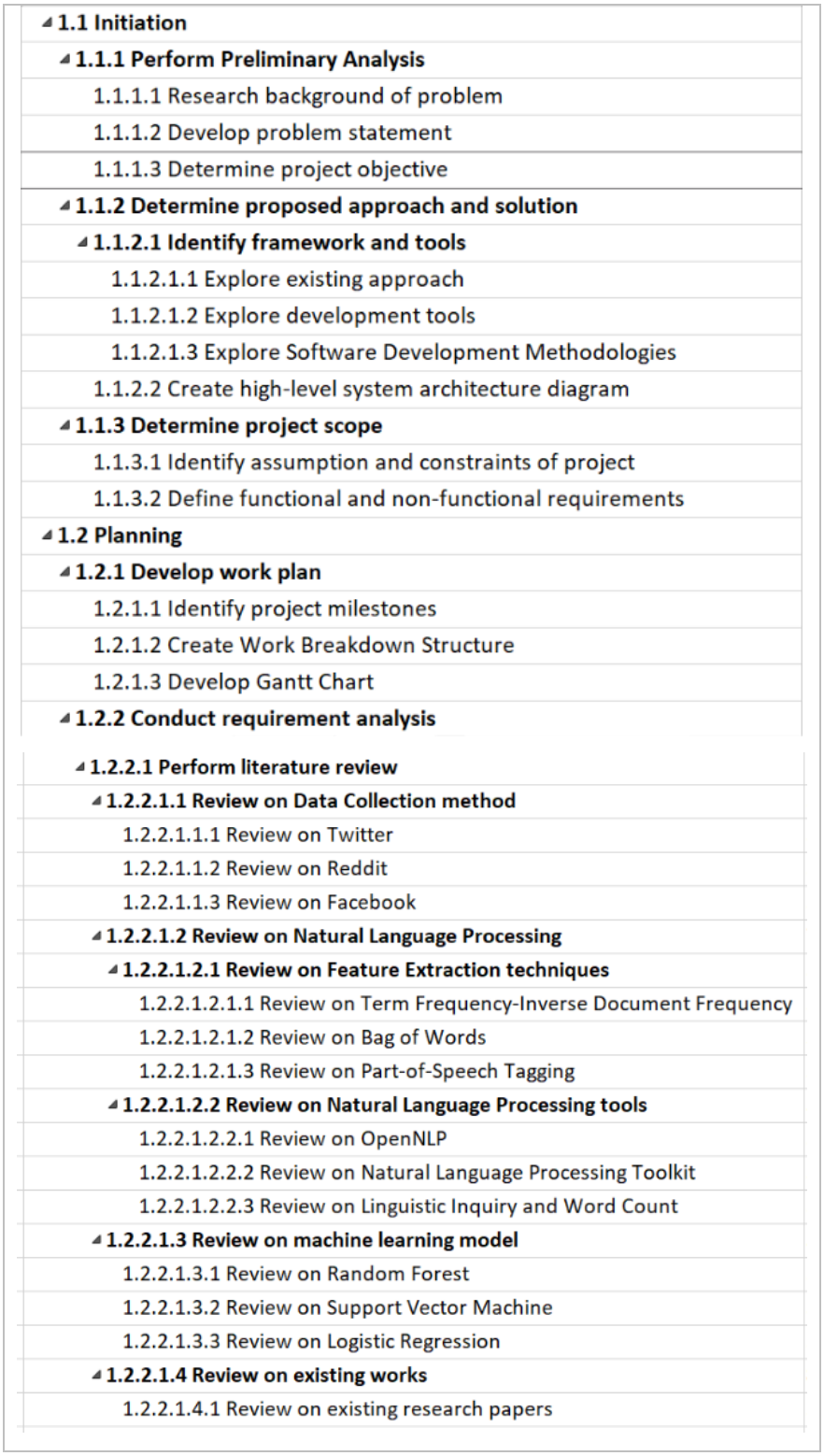


Figure 3.2: Work Breakdown Structure Part 1

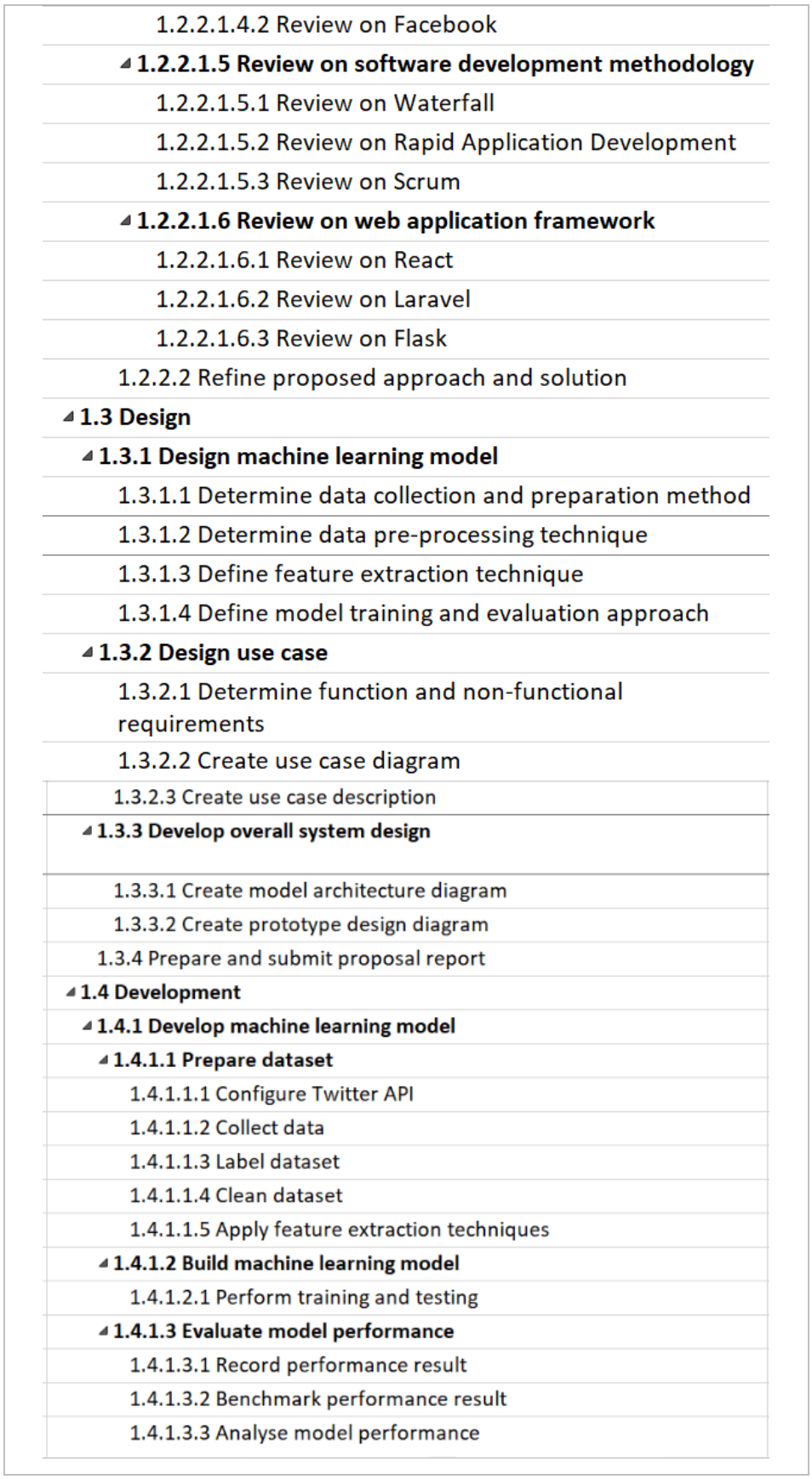


Figure 3.3: Work Breakdown Structure Part 2

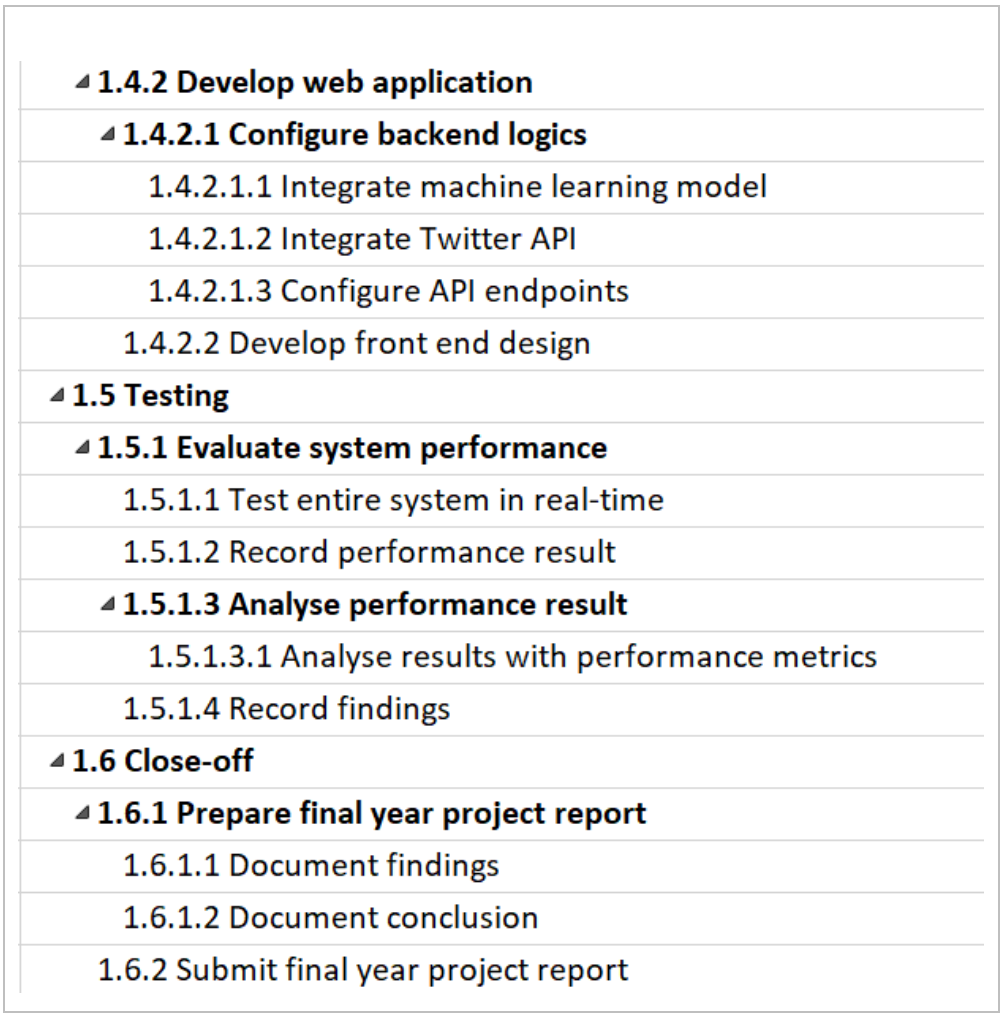
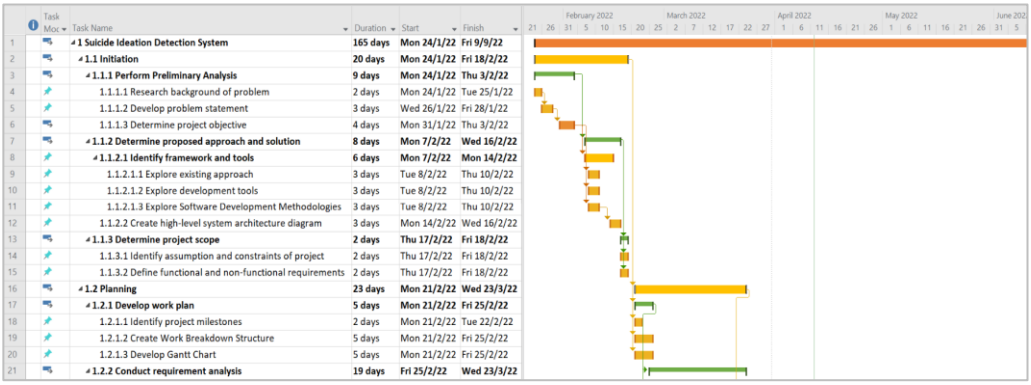


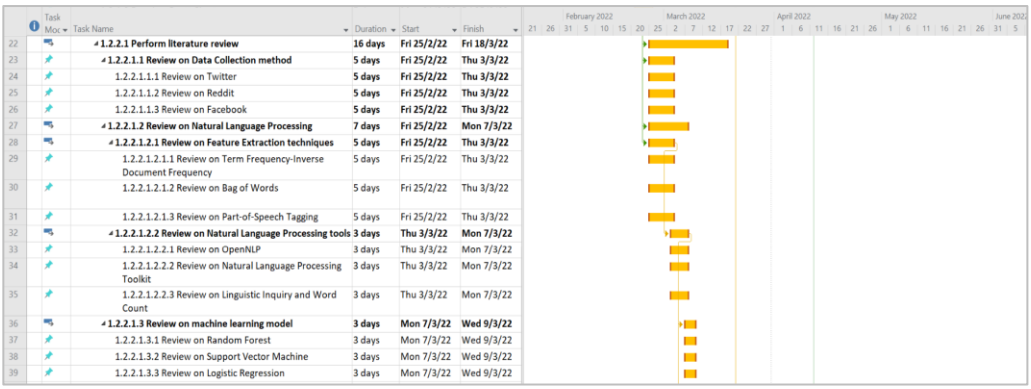
Figure 3.4: Work Breakdown Structure Part 3

3.4.2 Gantt Chart



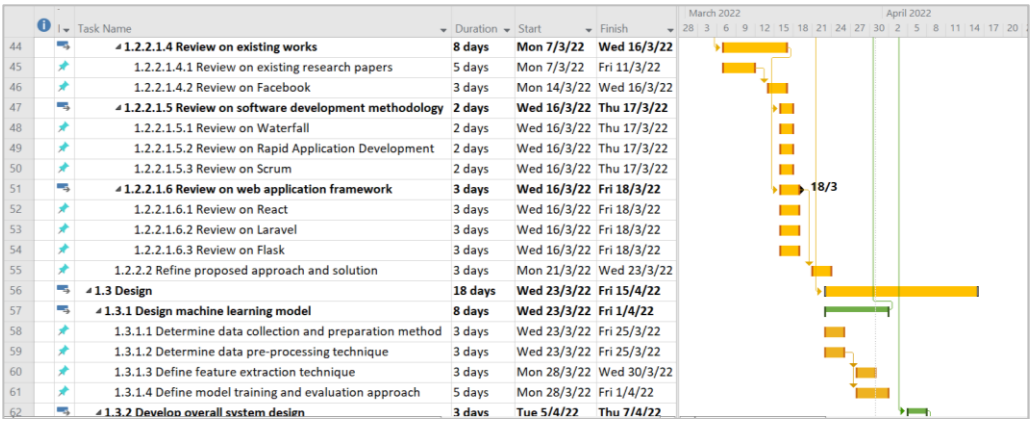
	Task Mode	Task Name	Duration	Start	Finish
1		**1 Suicide Ideation Detection System**	**165 days**	**Mon 24/1/22**	**Fri 9/9/22**
2		**1.1 Initiation**	**20 days**	**Mon 24/1/22**	**Fri 18/2/22**
3		**1.1.1 Perform Preliminary Analysis**	**9 days**	**Mon 24/1/22**	**Thu 3/2/22**
4		1.1.1.1 Research background of problem	2 days	Mon 24/1/22	Tue 25/1/22
5		1.1.1.2 Develop problem statement	3 days	Wed 26/1/22	Fri 28/1/22
6		1.1.1.3 Determine project objective	4 days	Mon 31/1/22	Thu 3/2/22
7		**1.1.2 Determine proposed approach and solution**	**8 days**	**Mon 7/2/22**	**Wed 16/2/22**
8		**1.1.2.1 Identify framework and tools**	**6 days**	**Mon 7/2/22**	**Mon 14/2/22**
9		1.1.2.1.1 Explore existing approach	3 days	Tue 8/2/22	Thu 10/2/22
10		1.1.2.1.2 Explore development tools	3 days	Tue 8/2/22	Thu 10/2/22
11		1.1.2.1.3 Explore Software Development Methodologies	3 days	Tue 8/2/22	Thu 10/2/22
12		1.1.2.2 Create high-level system architecture diagram	3 days	Mon 14/2/22	Wed 16/2/22
13		**1.1.3 Determine project scope**	**2 days**	**Thu 17/2/22**	**Fri 18/2/22**
14		1.1.3.1 Identify assumption and constraints of project	2 days	Thu 17/2/22	Fri 18/2/22
15		1.1.3.2 Define functional and non-functional requirements	2 days	Thu 17/2/22	Fri 18/2/22
16		**1.2 Planning**	**23 days**	**Mon 21/2/22**	**Wed 23/3/22**
17		**1.2.1 Develop work plan**	**5 days**	**Mon 21/2/22**	**Fri 25/2/22**
18		1.2.1.1 Identify project milestones	2 days	Mon 21/2/22	Tue 22/2/22
19		1.2.1.2 Create Work Breakdown Structure	5 days	Mon 21/2/22	Fri 25/2/22
20		1.2.1.3 Develop Gantt Chart	5 days	Mon 21/2/22	Fri 25/2/22
21		**1.2.2 Conduct requirement analysis**	**19 days**	**Fri 25/2/22**	**Wed 23/3/22**

Figure 3.5: Gantt Chart Part 1



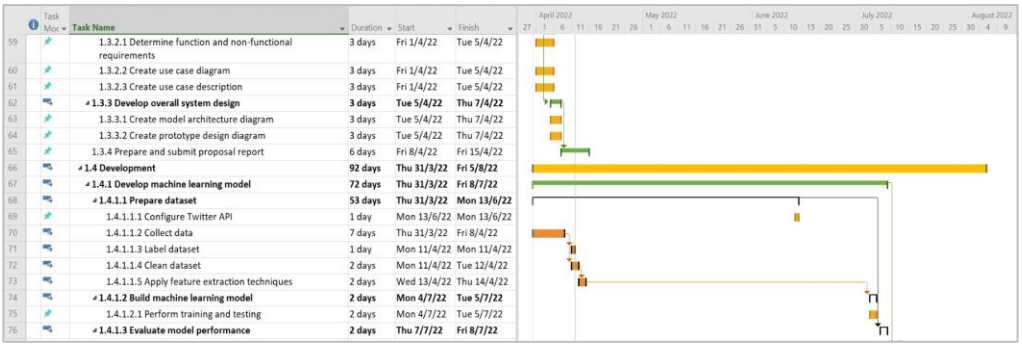
	Task Mode	Task Name	Duration	Start	Finish
22		**1.2.2.1 Perform literature review**	**16 days**	**Fri 25/2/22**	**Fri 18/3/22**
23		**1.2.2.1.1 Review on Data Collection method**	**5 days**	**Fri 25/2/22**	**Thu 3/3/22**
24		1.2.2.1.1.1 Review on Twitter	5 days	Fri 25/2/22	Thu 3/3/22
25		1.2.2.1.1.2 Review on Reddit	5 days	Fri 25/2/22	Thu 3/3/22
26		1.2.2.1.1.3 Review on Facebook	5 days	Fri 25/2/22	Thu 3/3/22
27		**1.2.2.1.2 Review on Natural Language Processing**	**7 days**	**Fri 25/2/22**	**Mon 7/3/22**
28		**1.2.2.1.2.1 Review on Feature Extraction techniques**	**5 days**	**Fri 25/2/22**	**Thu 3/3/22**
29		1.2.2.1.2.1.1 Review on Term Frequency-Inverse Document Frequency	5 days	Fri 25/2/22	Thu 3/3/22
30		1.2.2.1.2.1.2 Review on Bag of Words	5 days	Fri 25/2/22	Thu 3/3/22
31		1.2.2.1.2.1.3 Review on Part-of-Speech Tagging	5 days	Fri 25/2/22	Thu 3/3/22
32		**1.2.2.1.2.2 Review on Natural Language Processing tools**	**3 days**	**Thu 3/3/22**	**Mon 7/3/22**
33		1.2.2.1.2.2.1 Review on OpenNLP	3 days	Thu 3/3/22	Mon 7/3/22
34		1.2.2.1.2.2.2 Review on Natural Language Processing Toolkit	3 days	Thu 3/3/22	Mon 7/3/22
35		1.2.2.1.2.2.3 Review on Linguistic Inquiry and Word Count	3 days	Thu 3/3/22	Mon 7/3/22
36		**1.2.2.1.3 Review on machine learning model**	**3 days**	**Mon 7/3/22**	**Wed 9/3/22**
37		1.2.2.1.3.1 Review on Random Forest	3 days	Mon 7/3/22	Wed 9/3/22
38		1.2.2.1.3.2 Review on Support Vector Machine	3 days	Mon 7/3/22	Wed 9/3/22
39		1.2.2.1.3.3 Review on Logistic Regression	3 days	Mon 7/3/22	Wed 9/3/22

Figure 3.6: Gantt Chart Part 2



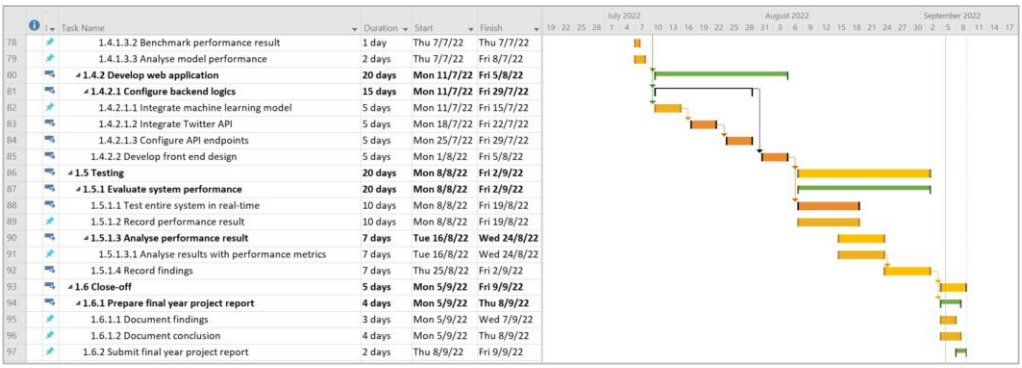
	Task Mode	Task Name	Duration	Start	Finish
44		**1.2.2.1.4 Review on existing works**	**8 days**	**Mon 7/3/22**	**Wed 16/3/22**
45		1.2.2.1.4.1 Review on existing research papers	5 days	Mon 7/3/22	Fri 11/3/22
46		1.2.2.1.4.2 Review on Facebook	3 days	Mon 14/3/22	Wed 16/3/22
47		**1.2.2.1.5 Review on software development methodology**	**2 days**	**Wed 16/3/22**	**Thu 17/3/22**
48		1.2.2.1.5.1 Review on Waterfall	2 days	Wed 16/3/22	Thu 17/3/22
49		1.2.2.1.5.2 Review on Rapid Application Development	2 days	Wed 16/3/22	Thu 17/3/22
50		1.2.2.1.5.3 Review on Scrum	2 days	Wed 16/3/22	Thu 17/3/22
51		**1.2.2.1.6 Review on web application framework**	**3 days**	**Wed 16/3/22**	**Fri 18/3/22**
52		1.2.2.1.6.1 Review on React	3 days	Wed 16/3/22	Fri 18/3/22
53		1.2.2.1.6.2 Review on Laravel	3 days	Wed 16/3/22	Fri 18/3/22
54		1.2.2.1.6.3 Review on Flask	3 days	Wed 16/3/22	Fri 18/3/22
55		1.2.2.2 Refine proposed approach and solution	3 days	Mon 21/3/22	Wed 23/3/22
56		**1.3 Design**	**18 days**	**Wed 23/3/22**	**Fri 15/4/22**
57		**1.3.1 Design machine learning model**	**8 days**	**Wed 23/3/22**	**Fri 1/4/22**
58		1.3.1.1 Determine data collection and preparation method	3 days	Wed 23/3/22	Fri 25/3/22
59		1.3.1.2 Determine data pre-processing technique	3 days	Wed 23/3/22	Fri 25/3/22
60		1.3.1.3 Define feature extraction technique	3 days	Mon 28/3/22	Wed 30/3/22
61		1.3.1.4 Define model training and evaluation approach	5 days	Mon 28/3/22	Fri 1/4/22
62		**1.3.2 Develop overall system design**	**3 days**	**Tue 5/4/22**	**Thu 7/4/22**

Figure 3.7: Gantt Chart Part 3



	Task Mode	Task Name	Duration	Start	Finish
59		1.3.2.1 Determine function and non-functional requirements	3 days	Fri 1/4/22	Tue 5/4/22
60		1.3.2.2 Create use case diagram	3 days	Fri 1/4/22	Tue 5/4/22
61		1.3.2.3 Create use case description	3 days	Fri 1/4/22	Tue 5/4/22
62		**1.3.3 Develop overall system design**	**3 days**	**Tue 5/4/22**	**Thu 7/4/22**
63		1.3.3.1 Create model architecture diagram	3 days	Tue 5/4/22	Thu 7/4/22
64		1.3.3.2 Create prototype design diagram	3 days	Tue 5/4/22	Thu 7/4/22
65		1.3.4 Prepare and submit proposal report	6 days	Fri 8/4/22	Fri 15/4/22
66		**1.4 Development**	**92 days**	**Thu 31/3/22**	**Fri 5/8/22**
67		**1.4.1 Develop machine learning model**	**72 days**	**Thu 31/3/22**	**Fri 8/7/22**
68		**1.4.1.1 Prepare dataset**	**53 days**	**Thu 31/3/22**	**Mon 13/6/22**
69		1.4.1.1.1 Configure Twitter API	1 day	Mon 13/6/22	Mon 13/6/22
70		1.4.1.1.2 Collect data	7 days	Thu 31/3/22	Fri 8/4/22
71		1.4.1.1.3 Label dataset	1 day	Mon 11/4/22	Mon 11/4/22
72		1.4.1.1.4 Clean dataset	2 days	Mon 11/4/22	Tue 12/4/22
73		1.4.1.1.5 Apply feature extraction techniques	2 days	Wed 13/4/22	Thu 14/4/22
74		**1.4.1.2 Build machine learning model**	**2 days**	**Mon 4/7/22**	**Tue 5/7/22**
75		1.4.1.2.1 Perform training and testing	2 days	Mon 4/7/22	Tue 5/7/22
76		**1.4.1.3 Evaluate model performance**	**2 days**	**Thu 7/7/22**	**Fri 8/7/22**

Figure 3.8: Gantt Chart Part 4



		Task Name	Duration	Start	Finish
78		1.4.1.3.2 Benchmark performance result	1 day	Thu 7/7/22	Thu 7/7/22
79		1.4.1.3.3 Analyse model performance	2 days	Thu 7/7/22	Fri 8/7/22
80		**1.4.2 Develop web application**	**20 days**	**Mon 11/7/22**	**Fri 5/8/22**
81		**1.4.2.1 Configure backend logics**	**15 days**	**Mon 11/7/22**	**Fri 29/7/22**
82		1.4.2.1.1 Integrate machine learning model	5 days	Mon 11/7/22	Fri 15/7/22
83		1.4.2.1.2 Integrate Twitter API	5 days	Mon 18/7/22	Fri 22/7/22
84		1.4.2.1.3 Configure API endpoints	5 days	Mon 25/7/22	Fri 29/7/22
85		1.4.2.2 Develop front end design	5 days	Mon 1/8/22	Fri 5/8/22
86		**1.5 Testing**	**20 days**	**Mon 8/8/22**	**Fri 2/9/22**
87		**1.5.1 Evaluate system performance**	**20 days**	**Mon 8/8/22**	**Fri 2/9/22**
88		1.5.1.1 Test entire system in real-time	10 days	Mon 8/8/22	Fri 19/8/22
89		1.5.1.2 Record performance result	10 days	Mon 8/8/22	Fri 19/8/22
90		**1.5.1.3 Analyse performance result**	**7 days**	**Tue 16/8/22**	**Wed 24/8/22**
91		1.5.1.3.1 Analyse results with performance metrics	7 days	Tue 16/8/22	Wed 24/8/22
92		1.5.1.4 Record findings	7 days	Thu 25/8/22	Fri 2/9/22
93		**1.6 Close-off**	**5 days**	**Mon 5/9/22**	**Fri 9/9/22**
94		**1.6.1 Prepare final year project report**	**4 days**	**Mon 5/9/22**	**Thu 8/9/22**
95		1.6.1.1 Document findings	3 days	Mon 5/9/22	Wed 7/9/22
96		1.6.1.2 Document conclusion	4 days	Mon 5/9/22	Thu 8/9/22
97		1.6.2 Submit final year project report	2 days	Thu 8/9/22	Fri 9/9/22

Figure 3.9: Gantt Chart Part 5

CHAPTER 4

PROJECT SPECIFICATIONS

4.1 Introduction

This section introduces the general design of the machine learning model's architecture with detailed description on how the textual data was captured, analysed and passed into the model for prediction. This section also specifies the functional and non-functional requirements of the project, as well as the prototype design of the web application.

4.2 Machine Learning Model

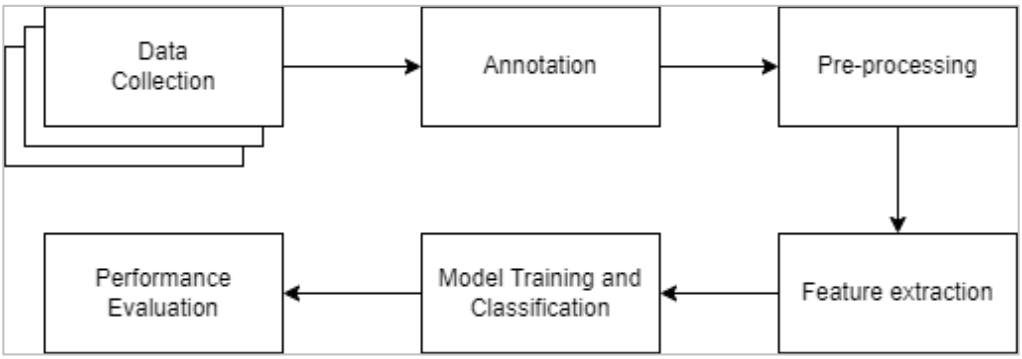


Figure 4.1: Overview on Machine Learning Model Development

Figure 4.1 illustrates the overall flow of the machine learning model development. It begins with collecting the tweets from Twitter through its API. Then, the tweets were manually annotated according to its associated suicide risk level. From there, the tweets were pre-processed to clean and reduce noisy data. Thereafter, feature extraction techniques were applied to extract the most significant features of each tweet, in order to pass into the machine learning model, Random Forest for its classification task. Finally, the performance of the model was evaluated and benchmarked until it reaches its desired performance result.

4.2.1 Data Collection and Preparation

Due to data and privacy concerns, there are little to no publicly available datasets on tweets related to suicide. Therefore, the data was manually collected from the historical tweets through Twitter API to build the ground-truth data that will be used to train the classification model.

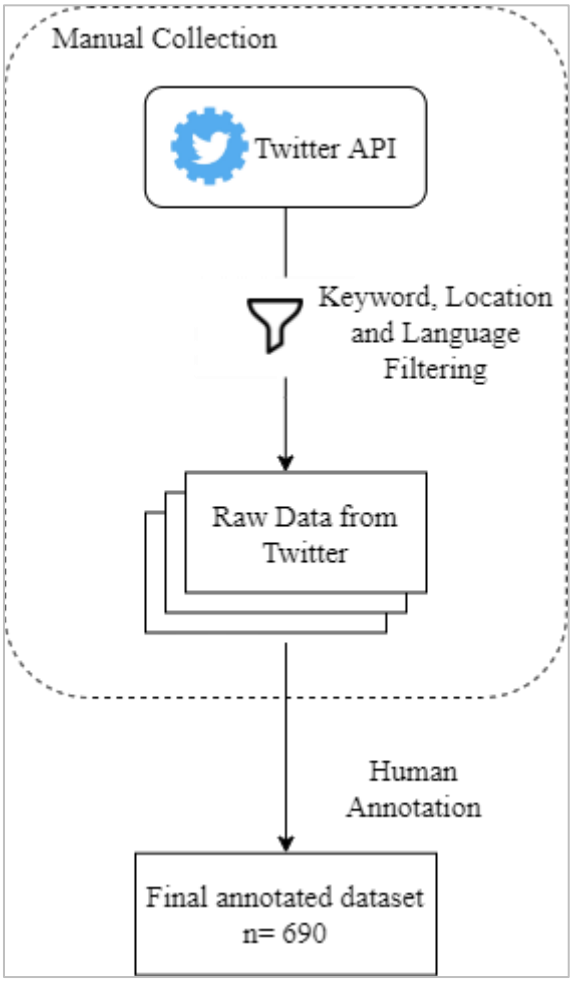


Figure 4.2: Overview on Data Collection and Preparation

Figure 4.2 shows the overview on the data collection and preparation method used in this project. Firstly, the filter was set to collect tweets from the United States and in English language only. From there, keywords filtering approach was used to capture the tweets based on words or phrases that are associated with suicide, as validated by the works of Parrott et al. (2020). The keyword filter was based on terms that are validated by more than 80% of the respondents in their research works, as shown in Figure 4.3. Hence, a total of

22 keywords were used for the filter, which include terms such as “better off dead”, “suicide”, “slit my wrists” and “suicidal ideation”.

Term	Yes, people would use this term	No, people would not
better off dead	73	1
suicide	72	1
slit my wrists	72	2
suicidal ideation	71	3
blow my brains out	71	3
hang myself	71	2
end it	71	3
shoot myself	70	4
suicidal	70	3
self-harm	70	4
suicidal behaviors	69	5
murder-suicide	68	6
fleeting thoughts of suicide	68	5
there's nothing left for me	67	7
completed suicide	64	10
blow my head off	64	10
off myself	63	11
committed suicide	62	12
suicide intent	62	11
do myself in	61	13
cry for help	61	13
failed attempt	59	15

Figure 4.3: Suicide-indicative Terms (Parrott et al., 2020)

The extracted tweets, along with its associated profile information such as geolocation, date and time of posting were stored in a csv file. Then, the tweets undergo the data annotation process, which is to assess and assign the tweets based on its associated level of suicide risk. Table 4.1 lists the risk annotation guidelines adapted from Shing et al. (2018).

Table 4.1: Risk Classification

Risk Label	**Description**
0 (Low Risk)	No evidence or patterns to suggest that the user is at risk of suicide
1 (Medium Risk)	Possible risk of suicide identified from the user content; no emergency aid required
2 (High Risk)	Strong and conclusive phrases used to display serious suicidal intent; emergency aid urgently required

4.2.2 Data Pre-processing

From the prepared dataset, the tweets undergo various pre-processing steps mainly by using Python's NLTK library. Figure 4.4 depicts the overview of the pre-processing steps that was performed on the dataset.

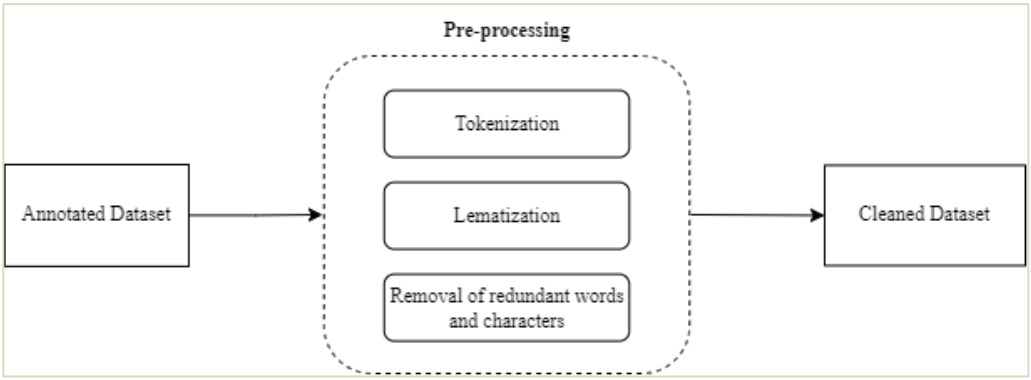


Figure 4.4: Data Pre-processing Method

The pre-processing tasks includes:

- Tokenization: to filter and convert the sentences in each tweet to individual words, referred to as tokens. These tokens help in ensuring that the input data can be efficiently processed during the machine learning process.
- Lemmatization: to normalize the words by mapping it back to its lemma. This is to reduce the dimensionality of the data, which helps in easing the identification of synonyms during the sentiment analysis process.
- Removal of redundant data: to exclude meaningless data such as stop words, special characters, brackets and URLs. This will help in reducing the noise of the data that can lead to better performance of the classifier.

4.2.3 Feature Extraction

Once the dataset is cleaned, feature extraction techniques were applied to build the feature set that will be passed into the classifier for its classification task.

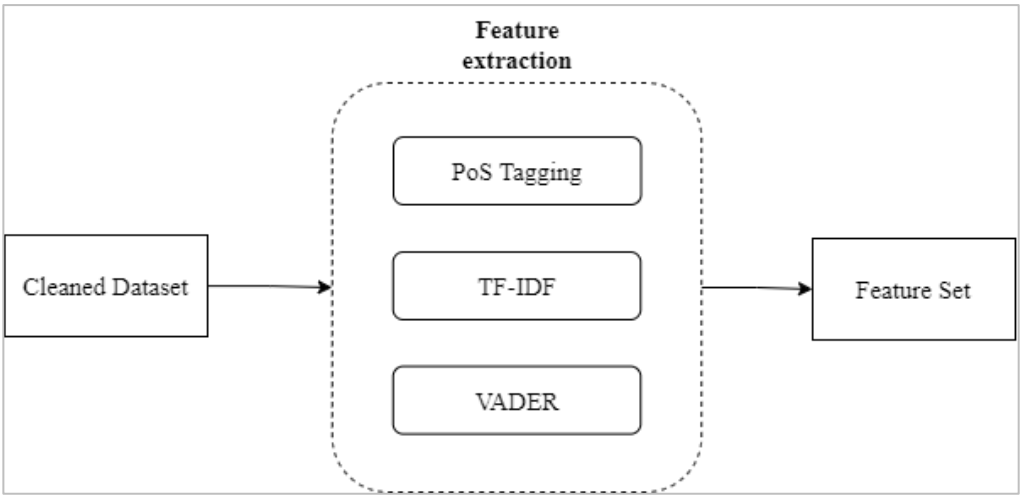


Figure 4.5: Feature Extraction

Figure 4.5 depicts the approach used to build the feature sets which consists of word frequency, syntactic and sentiment features.

4.2.3.1 Word Frequency

TF-IDF was used to evaluate terms that contain critical information within the tweets in the dataset. The feature set was built from the output of the TF-IDF function, where irrelevant terms are assigned with a low score, while the terms that represent important semantic information are given a higher score. This feature set helps the model to determine the suicide risk level, as the model analyses each tweet based on the absence or presence of specific terms within the tweet. Given its ability to analyse large textual datasets within a short period of time, the use of TF-IDF optimizes the training time of the classification model.

4.2.3.2 Syntactic Feature

PoS tagging was used to obtain the syntactic properties derived from the parts of speech in each tweet, which enhances the ability of the model to detect the similarities between the tweets. The PoS tags provides context on the grammatical subgroups such as verbs, nouns and adjectives each word of the sentence belongs to. From there, the total number of words in each grammatical category was computed.

4.2.3.3 Sentiment Feature

Sentiment analysis was performed to analyse and extract the emotional sentiment associated with each tweet by using NLTK's VADER. VADER analyses and assigns the sentiment score based on the intensity of each sentiment identified through each tweet. It works efficiently on social media data since it considers the use of punctuations, capital letters, Internet slangs and emoticons with effective identification of polarity shifts in sentences during the computation of the sentiment score. The sentiment score provides a preliminary understanding on the general sentiment expressed through the tweet, which will contribute towards the preparation of the final dataset, as well as the model evaluation.

4.2.4 Model Training

The hold out method was used for model training and testing in the project. The feature set was split into 80% training (552 samples) and 20% test sets (138 samples). The training set is used to train the model, while the testing set is used to evaluate the model's behaviour when unseen samples are used as an input to the model. The risk label was stratified to ensure that equal distribution of each class size is included in both the sets used for training and testing, to reduce the risk of model bias. The training sets was passed into the machine learning model, Random Forest for its classification task. Then, the performance metrics obtained from the test set was computed for the evaluation of the model performance.

4.2.5 Performance Evaluation

The results of the model performance were validated based on the metrics: precision, recall and accuracy, which are commonly used in the previous works. True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) are the four major parameters used in the formula of these metrics, and they are obtained from the Confusion Matrix, which maps the information on the actual and predicted class labels by the model. The equation for precision, accuracy and recall are defined as follows:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (4.1)$$

$$Precision = \frac{TP}{TP+FP} \quad (4.2)$$

$$Recall = \frac{TP}{TP+FN} \quad (4.3)$$

where

TP= True Positive

TN= True Negative

FP= False Positive

FN= False Negative

Accuracy computes the ratio of correctly labelled risk levels to the total number of instances included in the predictions. However, accuracy alone is not sufficient to measure and evaluate the model performance, especially since accuracy is subjected to class imbalance bias. Therefore, metrics that focuses on the class labels such as precision will be used to compute the percentage of of correctly labelled risk levels. This will give an indication on how often the model correctly predict each risk level. Besides that, recall will also be used to compute the proportion of actual true positives that predicted correctly by the model.

In addition to these 3 metrics, the discrepancy rate was also used to gain insight on the percentage of misclassified samples. The calculation for the discrepancy rate is as follows:

$$Discrepancy\ Rate = 1 - Accuracy \quad (4.4)$$

where

Accuracy= Computed from Equation 4.1

If there is more than 15% of discrepancies found, the machine learning model should be further tuned to reach an output below this threshold.

4.3 System Requirements

4.3.1 Functional Requirements

1. The system shall allow the moderators to view the dashboard
2. The system shall allow the moderators to view the real-time statistics of the analysed tweets
3. The system shall allow the moderators to select an action for each flagged tweet
4. The system shall allow the moderators to update the action for each flagged tweet

4.3.2 Non-functional Requirements

1. Performance
    a. The system shall ensure that the accuracy of the Random Forest model must be at least 85%
    b. The system shall ensure that the precision of the Random Forest model must be at least 85%
    c. The system shall ensure that the recall of the Random Forest model must be at least 85%
2. Usability
    a. The system shall render the tweets flagged as high risk of suicide according to its descending order of tweet ID
    b. The system shall automatically send a direct message with mental health sources to individuals detected with medium or high-risk tweets
3. Reliability
    a. The system shall be able to handle real-time analysis and processing of Tweets with discrepancy rate below 15%
4. Flexibility
    a. The system shall enable the application to be adapted later for other social media applications
5. Security
    a. The system must adhere to the Twitter API Developer Policy

4.4 Use Case

4.4.1 Use Case Diagram

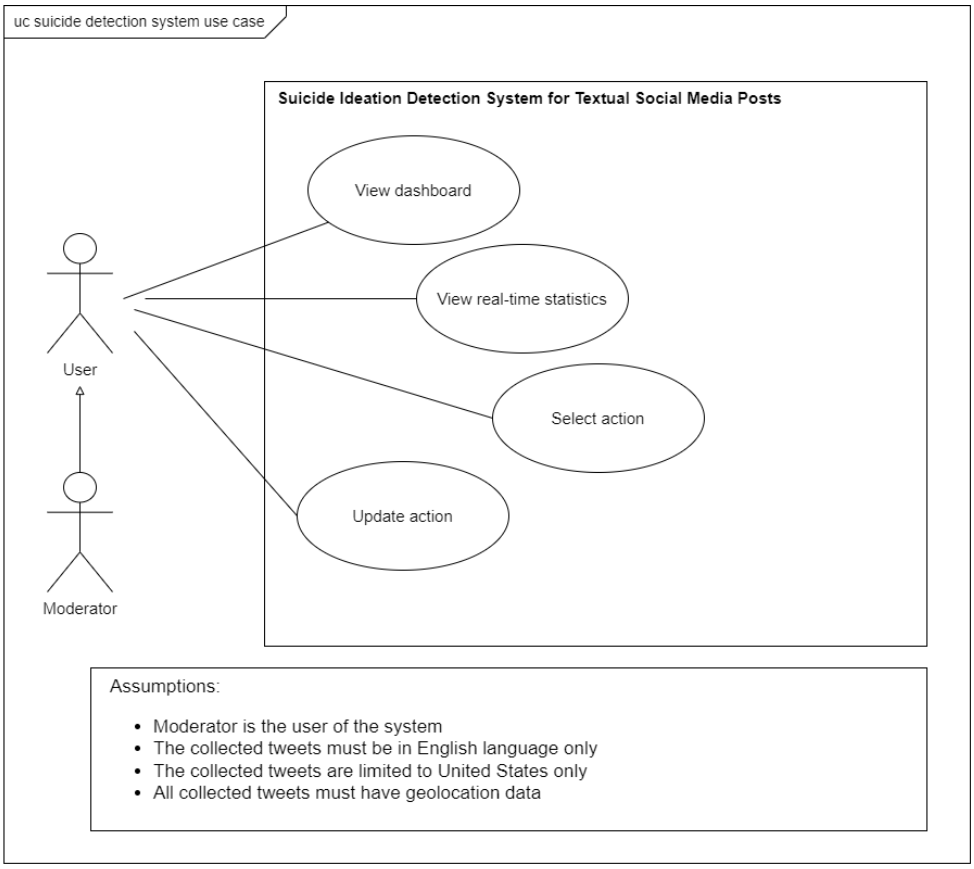


Figure 4.6: Use Case Diagram

4.4.2 Use Case Description

Use Case Name: View Dashboard	ID: 1	Importance Level: High
Primary Actor: User	Use Case Type: Detail and Essential	
Stakeholders and Interests: User wants to view the dashboard of the web application		
Brief Description: This use case mentions how the user can view the dashboard of the application, which contains details of the flagged and reviewed tweets.		
Trigger: User wishes to view the detailed content of the flagged and reviewed tweets.		
Relationships: Association: User Include: N/A Extend: Select Action, Update Action Generalization: N/A		
Normal Flow of Events: 1. User enters the main page 2. The system will display 2 tabs titled “Pending Review” and “Reviewed” 3. If user clicks on “Pending Review”, S-1: Display all flagged tweets is performed If user clicks on “Reviewed”, S-2: Display all reviewed tweets is performed		
SubFlows: S-1: Display all flagged tweets 1. System displays all high-risk tweets pending for review sorted according to the latest to oldest tweet, with details such as ID, content, author, date &time and location of tweet 2. If user clicks on twitter profile name under the “Author” column, the user is redirected to twitter profile 3. If user clicks on “Select Action” button, the system shall perform Use Case 3. S-2: Display all tweets 1. System displays all tweets that have been reviewed analysed by the system according to latest to oldest tweet, with details such as ID, content, author, date &time and location of tweet 2. If user clicks on twitter profile name under the “Author” column, the user is redirected to twitter profile 3. If use clicks on “Update Action” button, the system shall perform Use Case 4.		
Alternate/Exceptional Flows: -		

<table>
<tr><td colspan="2">Use Case Name: View Real-time Statistics</td><td>ID: 2</td><td>Importance Level: High</td></tr>
<tr><td colspan="2">Primary Actor: User</td><td colspan="2">Use Case Type: Detail and Essential</td></tr>
<tr><td colspan="4">Stakeholders and Interests:
User wants to view the real-time analysis statistics</td></tr>
<tr><td colspan="4">Brief Description:
This use case mentions how the user can view the real-time statistics on the total number of low, medium and high risk tweets analysed by the system, which is presented in a doughnut chart and line graph.</td></tr>
<tr><td colspan="4">Trigger: User wishes to view the analysis statistics of the system</td></tr>
<tr><td colspan="4">Relationships:
Association: User
Include: N/A
Extend: N/A
Generalization: N/A</td></tr>
<tr><td colspan="4">Normal Flow of Events:
1. User enters the main page
2. The system will display a doughnut chart on the total number of low, medium and high risk tweets analysed.
3. The system will display a line graph of the trend of the risk labels across the analysed tweets.</td></tr>
<tr><td colspan="4">SubFlows:
-</td></tr>
<tr><td colspan="4">Alternate/Exceptional Flows:
-</td></tr>
</table>

<table>
<tr><td colspan="2">Use Case Name: Select Action</td><td>ID: 3</td><td>Importance Level: High</td></tr>
<tr><td colspan="2">Primary Actor: User</td><td colspan="2">Use Case Type: Detail and Essential</td></tr>
<tr><td colspan="4">Stakeholders and Interests:
User wants to select an action for the flagged tweet</td></tr>
<tr><td colspan="4">Brief Description:
This use case describes how the user can select an action for each flagged tweet</td></tr>
<tr><td colspan="4">Trigger: User wishes to select an action for the flagged tweets on the system</td></tr>
<tr><td colspan="4">Relationships:
Association: User
Include: N/A
Extend: N/A
Generalization: N/A</td></tr>
<tr><td colspan="4">Normal Flow of Events:
1. User enters the “Pending Review” tab.
2. When user clicks on the “Select Action” button, the system display a pop-up modal for user to select action.
3. If “Safe to Ignore” is selected,
S-1: “Display Safe to Ignore” feedback modal is performed.
If “Alert the Authorities” is selected,
S-2: Display “Alert the Authorities” feedback modal is performed.</td></tr>
<tr><td colspan="4">SubFlows:
S-1: “Safe to Ignore” feedback modal
1. System displays feedback that action was performed successfully.
2. System displays the default message sent to the user.
3. If user clicks on “Close” icon, system closes the pop-up modal

S-2: “Alert the Authorities” feedback modal
1. System displays feedback that action was performed successfully.
2. System displays detailed information on the Twitter user, content, GPS coordinates of tweet and local authorities’ hotline.
3. If user clicks on “Close” icon, system closes the pop-up modal</td></tr>
<tr><td colspan="4">Alternate/Exceptional Flows:
-</td></tr>
</table>

Use Case Name: Update Action	ID: 4	Importance Level: High
Primary Actor: User	Use Case Type: Detail and Essential	
Stakeholders and Interests: User wants to update the action for the reviewed tweet		
Brief Description: This use case describes how the user can update the action for each reviewed tweet.		
Trigger: User wishes to update an action for the reviewed tweets on the system		
Relationships: Association: User Include: N/A Extend: N/A Generalization: N/A		
Normal Flow of Events: 1. User clicks on the "View All Tweets" tab 2. System displays all the analysed tweets. 3. User locates tweet with "Update Action" button. 4. User clicks on "Update Action" button. 5. If "Safe to Ignore" is selected, S-1: Display "Safe to Ignore" feedback modal is performed. If "Alert the Authorities" is selected, S-2: Display "Alert the Authorities" feedback modal is performed.		
SubFlows: S-1: "Safe to Ignore" feedback modal 1. System displays feedback that action was performed successfully. 2. System displays the default message sent to the user. 3. If user clicks on "Close" icon, system closes the pop-up modal S-2: "Alert the Authorities" feedback modal 1. System displays feedback that action was performed successfully. 2. System displays detailed information on the Twitter user, content, GPS coordinates of tweet and local authorities' hotline. 3. If user clicks on "Close" icon, system closes the pop-up modal		
Alternate/Exceptional Flows: -		

4.5 Prototype Design

4.5.1 Flowchart

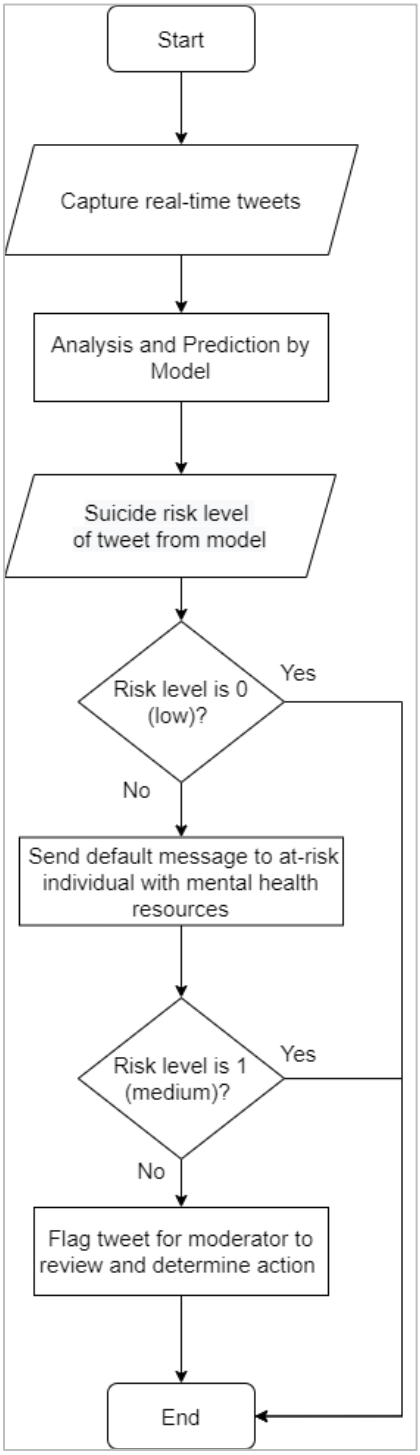


Figure 4.7: Flowchart for Web Application

4.5.2 Web Application Prototype

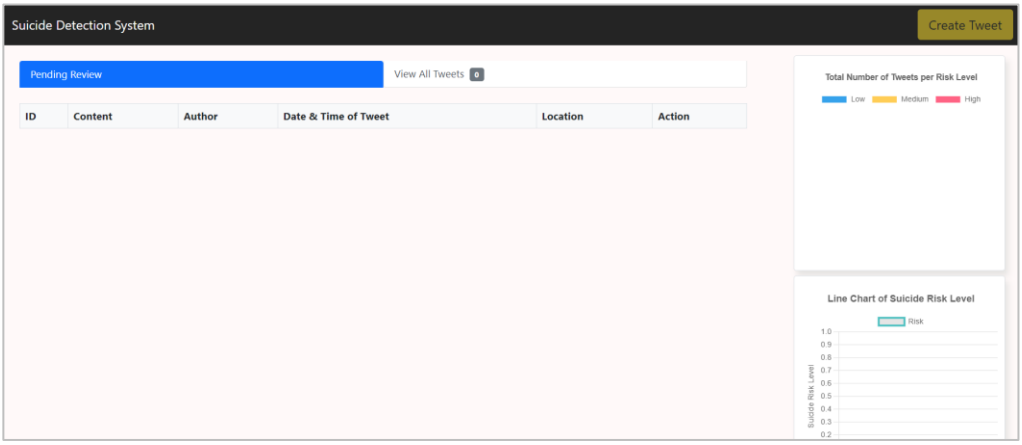


Figure 4.8: Web Application Prototype (Dashboard)

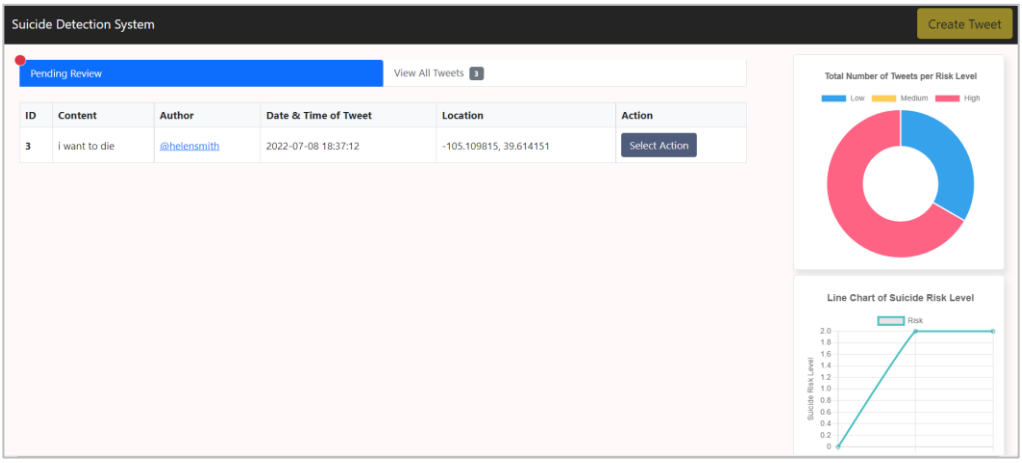


Figure 4.9: Web Application Prototype (Pending Review Tab)

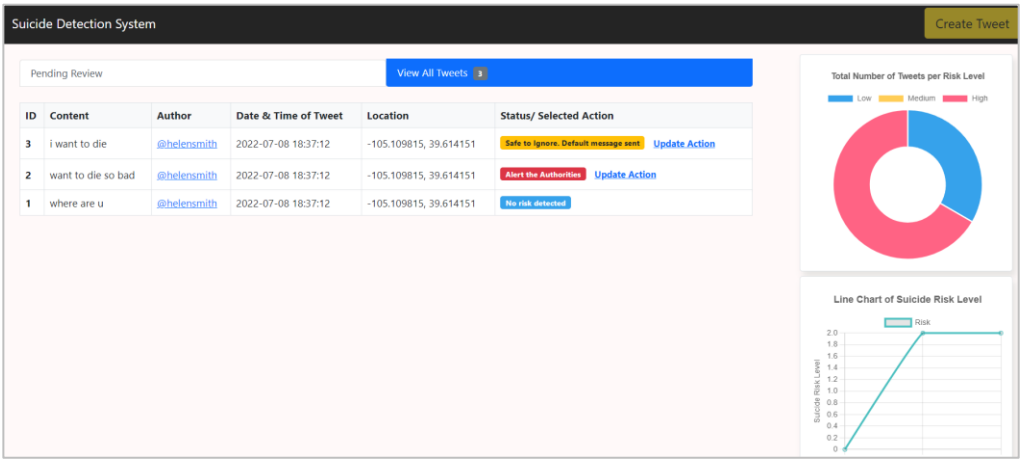


Figure 4.10: Web Application Prototype (View All Tweets Tab)

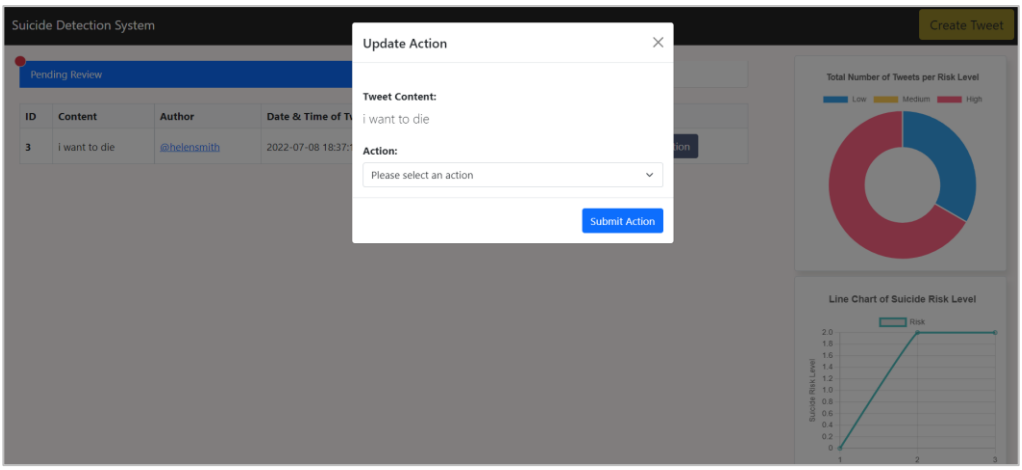


Figure 4.11: Web Application Prototype (Select Action Modal)

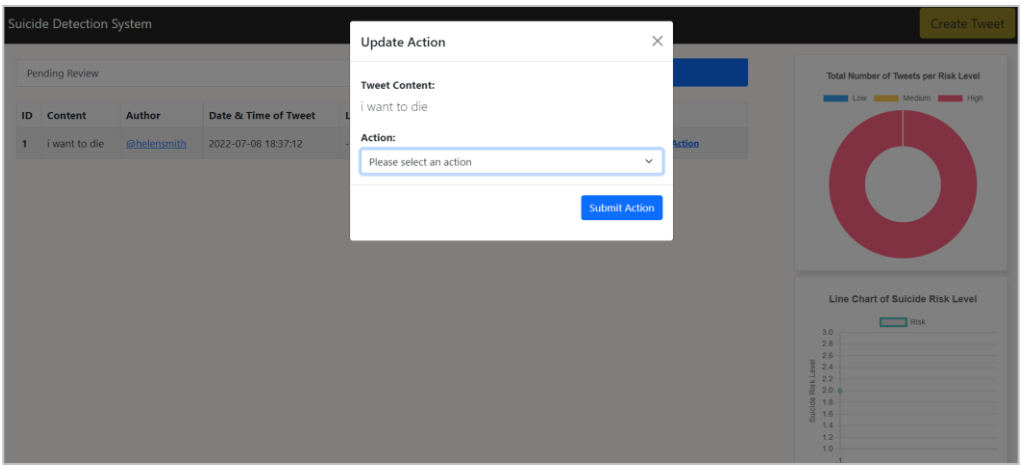


Figure 4.12: Web Application Prototype (Update Action Modal)

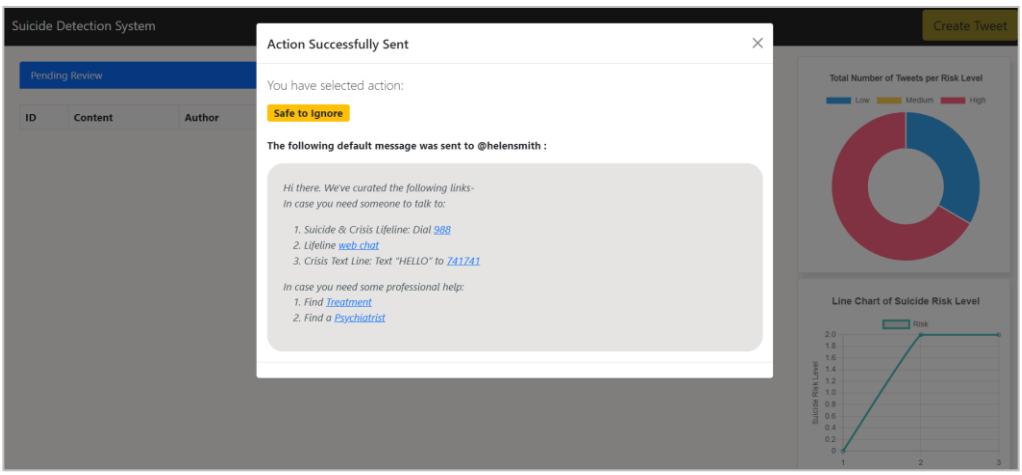


Figure 4.13: Web Application Prototype ("Safe to Ignore" Feedback Modal)

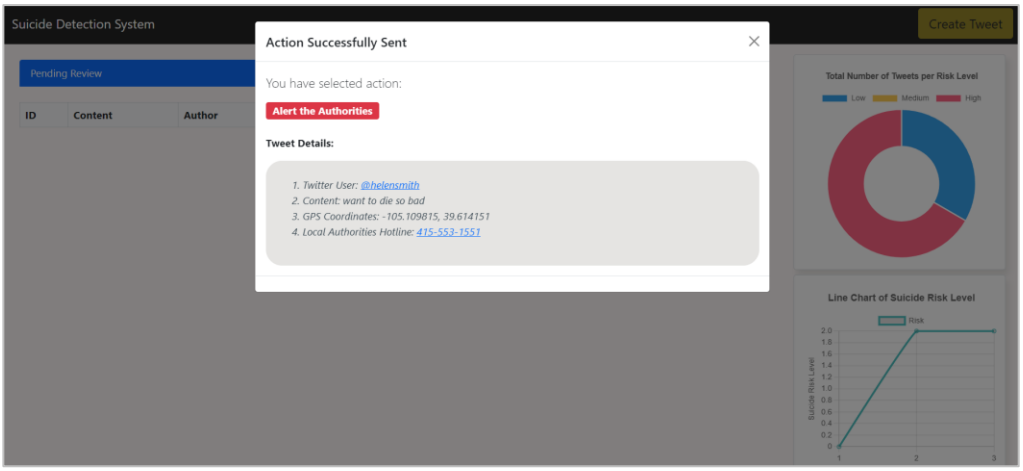


Figure 4.14: Web Application Prototype (“Alert the Authorities” Feedback Modal)

CHAPTER 5

DATA MODELLING

5.1 Data Collection

The dataset used in this project was built from scratch based on tweets that contain suicide indicative terms as listed in Figure 4.3. The basic assumption is that these terms would help in the collection of tweets that translates to different levels of suicide risk, which will be evaluated and labelled during the next stage.

```
import tweepy
import pandas as pd

consumer_key="omptQD9x8z61e074ACIkoiWdU"
consumer_secret="wCXW5fIsYlMj6IbRBa5ah1I0KxnGZlUAOVumDxa6PRUeZytYoV"
access_token="1434496584763461632-1VGNKOs0MH3cImh3cJwfGQ5GOgziBK"
access_token_secret="Fo15mTjyqpKcf1Ghy6g0JeOTbhnPN9GRiawsxd9NcP4Y9"

auth=tweepy.OAuthHandler(consumer_key,consumer_secret)
auth.set_access_token(access_token,access_token_secret)
api=tweepy.API(auth)
```

Figure 5.1: Twitter Authentication Code Snippet

Figure 5.1 shows the code snippet used to authenticate the Twitter API credentials. Tweepy, a Python library that facilitates the interactions with the Twitter API was utilised in order to automate the data collection process. As per Twitter API's security protocol, the requester's credentials are first authenticated using the access keys and tokens that are generated from the approved Twitter developer's account.

```
number_of_tweets=1000
full_text=[]
name=[]
screen_name=[]
date_time=[]
location_name=[]
coordinates=[]

for i in tweepy.Cursor(api.search_tweets,q="(suicidal thoughts) OR (contemplating suicide)-filter:retweets",lang="en",
                       geocode="37.0902,-95.7129,4500km",tweet_mode="extended").items(number_of_tweets):
    full_text.append(i.full_text)
    screen_name.append(i.user.screen_name)
    date_time.append(i.created_at)
    if(i.place is not None):
        location_name.append(i.place.full_name)
        coordinates.append(i.place.bounding_box.coordinates)
```

Figure 5.2: Data Collection Code Snippet

Once authorised, the tweets are collected in real-time based on a series of tokenised keywords that are specified in the query parameter of the "search_tweets" function. Figure 5.2 depicts the code snippet of the data collection functions, whereby additional filters are implemented to ensure that the tweets collected are from the United States and in English language only.

Initial analysis of the collected tweets revealed that while they do include suicide-indicative keywords, the majority of them show low risk of suicide. This could be attributed by the informal and concise nature of language and terms used on social media to express suicidal ideation. In contrast, terms that were passed into the query such as "fleeting thoughts of suicide" and "completed suicide" were more conventional among offline communication. Hence, these terms yielded many irrelevant results from the search function. As such, the list of suicide-indicative terms used for data collection was further refined based on the common terms and phrases identified from the initially collected tweets that do contain medium to high risk of suicide. The final list contains a total of a 16 suicide-indicative keywords, which are: *better off dead, suicide, slit my wrists, blow my brains out, hang myself, shoot myself, suicidal, blow my head off, suicidal, kill myself, want to die, suicidal thoughts, contemplating suicide, self harm, suicidal ideation and sleep forever*.

5.2 Data Annotation

The dataset was manually evaluated and annotated based on the risk category criteria defined previously in Table 4.1. Upon completion of the data annotation process, it was found that, although the phrases used in the search

query were refined in the previous section, the majority of the gathered tweets were still of low risk. The imbalanced distribution across the risk levels may have an impact on the performance of the classifier during model training. As the classifier is bias towards the majority class, it is more likely to misclassify the minority class, which in this case was the high-risk tweets. To mitigate this issue, additional low risk tweets were removed from the dataset. As such, the final annotated dataset consists of a total of 690 tweets, which is evenly distributed across each risk level (i.e.: 230 tweets per risk level). Table 5.1 illustrates an excerpt of the tweets based on each risk level:

Table 5.1: Excerpt of Tweets

Risk Label	**Example**
0 (Low Risk)	"… shoot some lovely pictures by myself today …"
1 (Medium Risk)	"… wish … shoot myself sometimes …"
2 (High Risk)	"…i want to kill myself … right now"

Further analysis on the dataset was conducted to outline the difference between the terms that are associated with medium and high suicide risk levels.

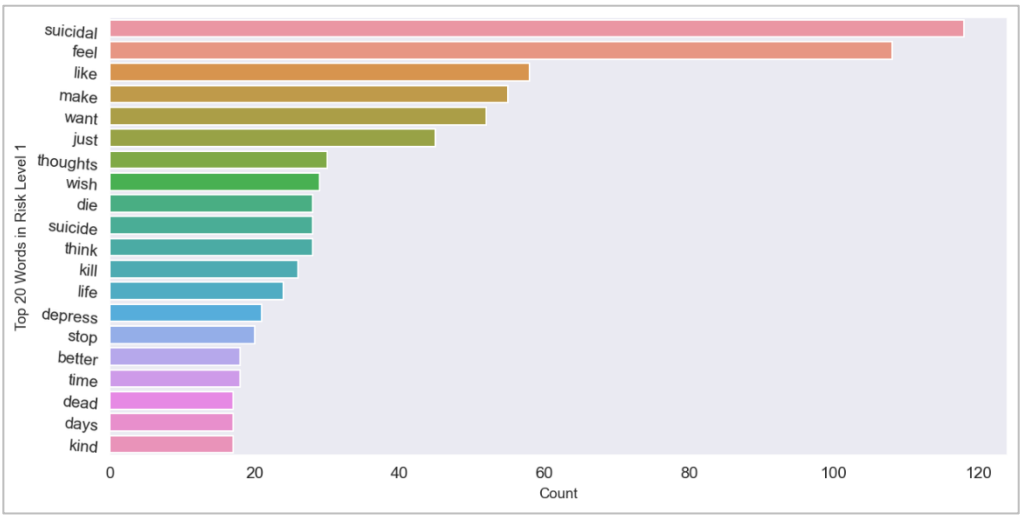


Figure 5.3: Top 20 Commonly Used Words in Medium Suicide Risk Level

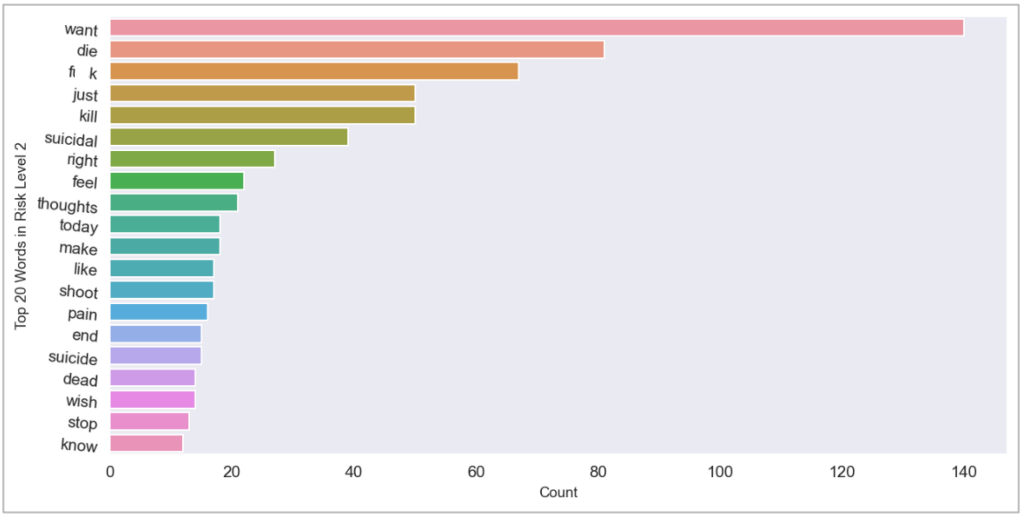


Figure 5.4: Top 20 Commonly Used Words used in High Suicide Risk Level

Based on Figure 5.3 and Figure 5.4, it is observed that while there are overlapping terms that are found in both of the risk levels, there is a significant difference in its use frequency. For example, the term “want” is most frequently found in tweets from high risk level, with a total occurrence of approximately 140 times. Although the same term is also identified in tweets from medium risk level, its total number of occurrences is less than 60, ranking 5 places behind dominant terms such as “suicidal”, “feel”, “like” and “make”.

This finding shows that tweets of medium risk levels generally consist of active phrases that indicate suicide ideation such as “suicidal”, “feel” and “thoughts”. High-risk tweets, on the other hand, are more proactive in nature, with the use explicit, strong and conclusive terms such as "want", "die" and “kill” suggesting behaviour associated with an imminent risk of suicide. This finding is consistent with the risk categorization criteria used during the data annotation process, in which tweets are labelled as medium risk if they induce the possibility of suicide, and high risk if the content displays substantial phrases pointing towards serious suicidal intent.

5.3 Data Pre-processing

Data pre-processing was carried out to clean and convert the dataset into the suitable format for the next stage.

5.3.1 Removal of Redundant Words and Characters

The dataset is passed into a custom function to remove redundant words and characters from each tweet. This includes username tags, links, punctuations, special characters, numbers and redundant whitespace, which are considered redundant as it does not contribute any value or meaning towards understanding the tweet. In addition to that, each contraction is broken down to its original group of words. As illustrated in Figure 5.5, this is to standardize the sentences to facilitate the grouping of the texts during the feature engineering stage, since not all terms are expressed in the form of contractions.

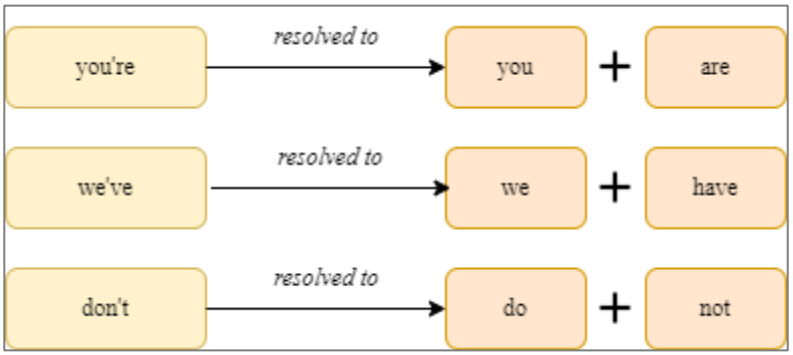


Figure 5.5: Standardization of Contractions

5.3.2 Tokenization and Lemmatization

Each tweet is first split into individual tokens by using NLTK's tokenizer. From there, the tokens are passed into the lemmatization function to analyse and normalise the words into its base form.

full_text	text_token	text_clean_token
work bout to make me start committing self harm again	['work', 'bout', 'to', 'make', 'me', 'start', 'committing', 'self', 'harm', 'again']	['work', 'bout', 'to', 'make', 'me', 'start', 'commit', 'self', 'harm', 'again']
fortnite sorry to everyone i was rude to today it is just i lost my girlfriend today and i want to commit suicide so i am not really in a good mood	['fortnite', 'sorry', 'to', 'everyone', 'i', 'was', 'rude', 'to', 'today', 'it', 'is', 'just', 'i', 'lost', 'my', 'girlfriend', 'today', 'and', 'i', 'want', 'to', 'commit', 'suicide', 'so', 'i',	['fortnite', 'sorry', 'to', 'everyone', 'i', 'be', 'rude', 'to', 'today', 'it', 'be', 'just', 'i', 'lose', 'my', 'girlfriend', 'today', 'and', 'i', 'want', 'to', 'commit', 'suicide', 'so', 'i', 'be', 'not', 'really', 'in', 'a', 'good', 'mood']
my father committed suicide when i was meher baba took over as my real father when i was happy fathers day beloved baba	'when', 'i', 'was', 'meher', 'baba', 'took', 'over', 'as', 'my', 'real', 'father', 'when', 'i', 'was', 'happy', 'fathers', 'day', 'beloved', 'baba']	['my', 'father', 'commit', 'suicide', 'when', 'i', 'be', 'meher', 'baba', 'take', 'over', 'as', 'my', 'real', 'father', 'when', 'i', 'be', 'happy', 'father', 'day', 'beloved', 'baba']

Figure 5.6: Snippet of Tokenized and Lemmatized Tweets

In this project, the lemmatization function emphasises on processing verbs, as it is observed that equivalent expressions of the same verb are represented across different words in the tweets. To illustrate, Figure 5.6 shows an excerpt on 3 tweet samples. From the *full_text* column, it is observed that the verb "commit" is expressed in 3 different grammatical tenses: "committing", "commit" and "committed". After passing the tokenised tweet into the lemmatization function, the *text_clean_token* column shows that these terms are now lemmatized to its based verb, "commit".

5.4 Feature Extraction

Once the data is cleaned, feature extraction is carried out to extract the features sets that will be passed into the machine learning model. Feature extraction seeks to improve model training performance by allowing the model to better correlate this information with its target class.

5.4.1 PoS Tagging

The PoS tags were extracted as features for the model to identify patterns within the grammatical properties that are associated with each tweet. Each tweet is first broken down into tokens, which are checked against 35 PoS subgroups and labelled according to its associated tag (Full description of the PoS tags annotations can be found in Appendix B, Table B-1).

```
[('i', 'NN'), ('consider', 'VBP'), ('go', 'VB'), ('ahead', 'RB'), ('and', 'CC'), ('jump', 'VB'), ('on', 'IN'), ('stream', 'N
N'), ('just', 'RB'), ('to', 'TO'), ('hang', 'VB'), ('out', 'RP'), ('but', 'CC'), ('tell', 'VB'), ('myself', 'PRP'), ('no', 'D
T'), ('just', 'RB'), ('decompress', 'NN')]
```

Figure 5.7: PoS Tags on Sample Tweet

To illustrate, Figure 5.7 shows the PoS tags labelled in accordance to its associated tokens for the full tweet that reads "I considered going ahead and jumping on stream just to hang out but told myself no, just decompress". The PoS tags represent the abbreviation of the part of speech category, which is defined internally by the NLTK library. Figure 5.8 shows an example of the feature set that corresponds to the sample tweet.

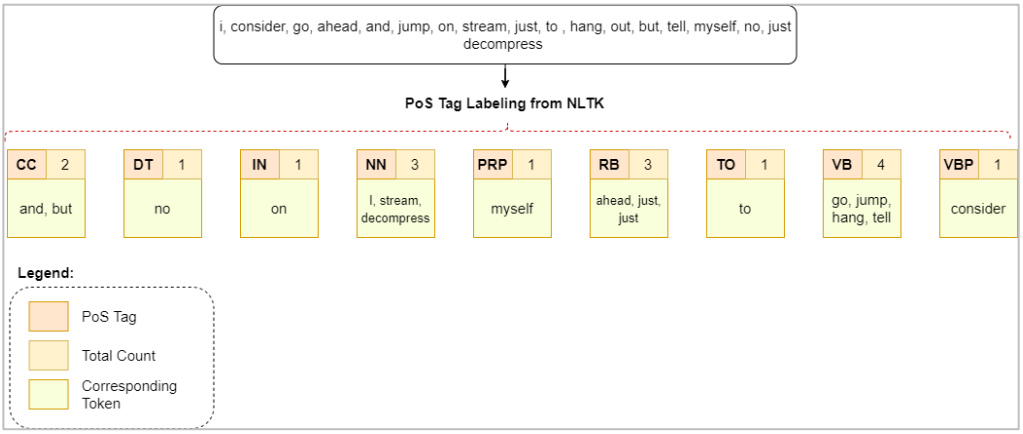


Figure 5.8: PoS Tag Feature Set for Sample Tweet

The total number of tokens categorised in each PoS tag subgroup were totalled to form the feature set, whereby the PoS tag represent the column name, while the total count represents its corresponding value. In Figure 5.8, the tokenised tweet was categorised across 10 PoS subgroups. The remaining 25 PoS subgroups that were not tagged with any tokens will be filled with 0.

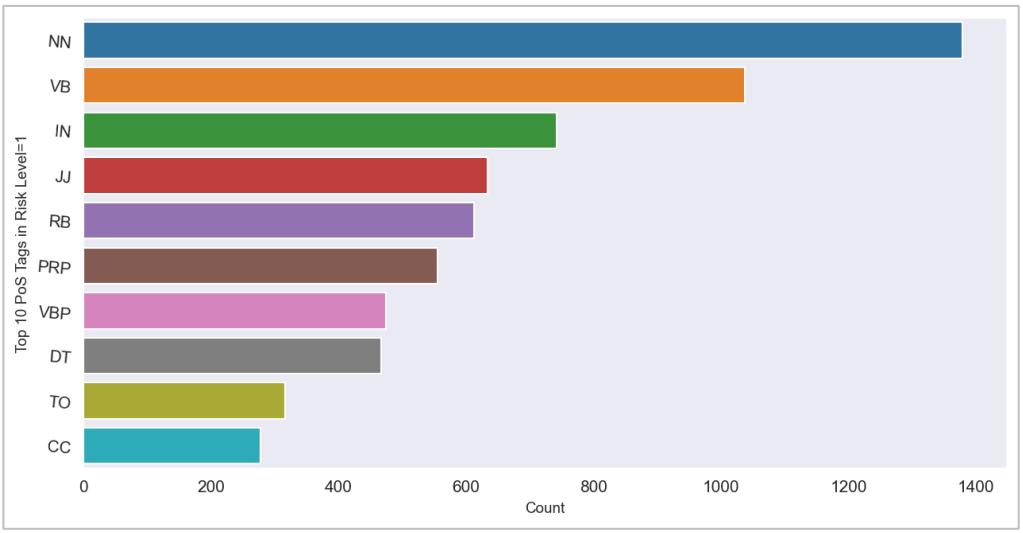


Figure 5.9: Top 10 PoS Tags for Medium Risk Tweets

Based on Figure 5.9 which shows the most tagged PoS labels within the tweets of low suicide risk, it is observed that singular noun (NN), base form verb (VB) and prepositions (IN) were common within this risk category.

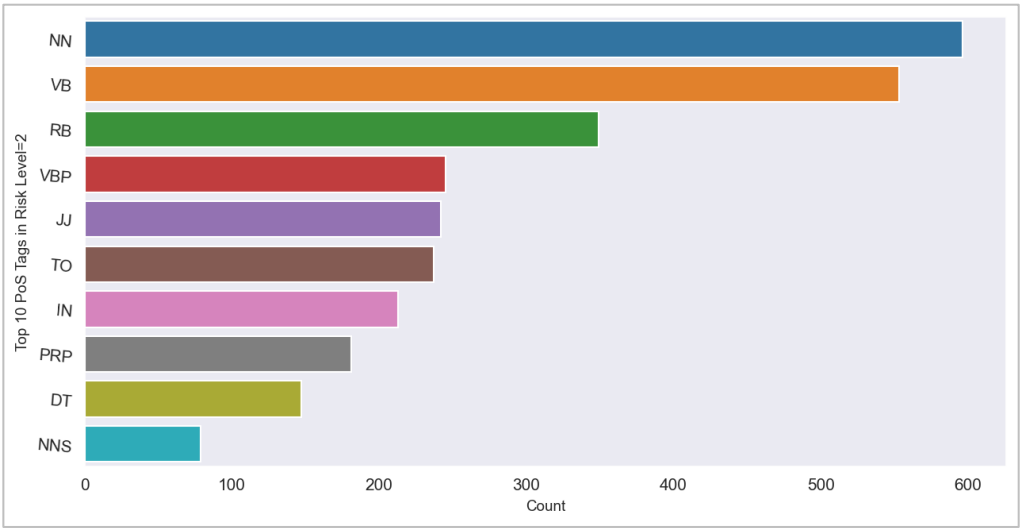


Figure 5.10: Top 10 PoS Tags for High Risk Tweets

On the other hand, Figure 5.10 shows the most tagged PoS labels within the tweets of medium suicide risk. It is observed that while overlaps exist between the tags in medium and high suicide risk, there is a significant difference in the total count, such that the count for nouns and based form verbs were lesser by half compared to medium risk tweets. Furthermore, adverbs (RB) and singular present verbs (VBP) were found to be significant within this risk category.

5.4.1.1 Summary

Through comparative analysis on the PoS tags, it was shown that the syntactic features associated with tweets of medium and high suicide risk differ slightly in terms of the frequency, as well as the grammatical properties used.

5.4.2 Sentiment Analysis

By using NLTK's VADER library, sentiment analysis was performed to understand the emotional sentiment associated with each tweet. By default, VADER produces 4 semantic scores for each input on its sentiment analyser function, this includes: negative, neutral, positive and compound score. The compound score is derived from normalizing the total of all semantic scores to the scale of -1 to 1. In general, the common threshold used to determine the

sentiment based on the compound score are defined as such: negative [-1, -0.05]; neutral (-0.05,0.05); positive [0.05,1].

5.4.2.1 Sample: Low Suicide Risk

Figure 5.11 illustrates the example output from VADER for a sample tweet taken from the dataset.

```
just happy to hang out with new cool and fun people instead of being by myself
###VADER Score Listing###
neg 0.0
neu 0.583
pos 0.417
compound 0.8519
```

Figure 5.11: VADER Score for Sample Tweet with Low Suicide Risk

Based on Figure 5.11, the output from VADER computed that the tweet was 58.3% neutral, 41.7% positive with a compound score of 85.19%. This is due to the choice of terms used and the structure of the sentence which consists of a combination of positive and neutral context. Furthermore, since there were no negative terms/ context expressed within the tweet, VADER computed a negative score of 0%.

5.4.2.2 Sample: Medium Suicide Risk

Figure 5.12 shows the output from VADER for a sample tweet of medium risk that is taken from the dataset.

```
sometimes i do feel like i am better off dead
###VADER Score Listing###
neg 0.257
neu 0.419
pos 0.323
compound 0.0258
```

Figure 5.12: VADER Score for Sample Tweet with Medium Suicide Risk

From Figure 5.12, it is observed that VADER computed that the tweet was 25.7% negative, 41.9% neutral, 32.3% positive with a compound score of 2.58%. Although the tweet is labelled as medium risk through human

annotation, the negative score on VADER is not as high when compared to the neutral and positive scores since the general context of the tweet includes a combination of different terms across the sentiment spectrum. Therefore, this yields a compound score that neither indicates that it is entirely positive nor negative.

However, there are some cases where contradiction exists, in which the compound score leans towards the positive end of the scale although the tweet is of medium suicide risk. An example is illustrated in Figure 5.13, where the combination of a high neutral and positive score led to a compound score of 55.49%.

```
i have a few not so bad days i think how long will this take until i slit my wrists
###VADER Score Listing###
neg 0.0
neu 0.841
pos 0.159
compound 0.5549
```

Figure 5.13: VADER Score for Sample Tweet with Medium Suicide Risk

This is attributed to the choice of words in used in the tweet, where the author used a combination of terms that were neither excessively optimistic nor pessimistic to convey their suicide ideation.

5.4.2.3 Sample: High Suicide Risk

```
i want to die
###VADER Score Listing###
neg 0.542
neu 0.278
pos 0.181
compound -0.5574
```

Figure 5.14: VADER Score for Sample Tweet with High Suicide Risk

Figure 5.14 shows the output from VADER for a sample tweet of high risk that is taken from the dataset. It is observed that VADER computed that the

tweet was 54.2% negative, 27.8% neutral, 18.1% positive, yielding a compound score of -55.74%.

However, alike the contradiction highlighted previously in section 5.4.2.2, there are cases where different combination of words used within the tweet results in a high compound score, although the general context of the tweet indicates that the user is in imminent risk of suicide. An example is illustrated in Figure 5.15.

```
someone do me a favor and blow my brains out right now
###VADER Score Listing###
neg 0.0
neu 0.803
pos 0.197
compound 0.4019
```

Figure 5.15: VADER Score for Sample Tweet with High Suicide Risk

5.4.2.4 Summary

Based on the findings gathered above, it is concluded that the sentiment scores from VADER is influenced by the choice of words used within the text. In other words, the sentiment score is dependent on the language intensity of the terms used. As such, the output from VADER alone is not always effective in assigning the appropriate sentiment score that accurately corresponds to the general context conveyed through the text. On that note, the threshold mentioned in section 5.4.2 should only be used as a guideline or reference on the overall sentiment of the input text.

5.4.3 TF-IDF

TF-IDF vectorizer is applied using unigram approach to transform each tweet into its vector representation. The vectorizer is configured to exclude exceptionally rare terms that appear in less than 5 tweets. For instance, if the term “love” appears in less than 5 tweets from the entire dataset, then it will not be taken into consideration during the computation of the weighted scores. Besides that, the maximum columns that can be produced from the vectorizer is set to 5000 to accommodate more frequented terms that can be found

through the tweets. From there, a word matrix of 278 columns is created, with each word labelled with the score in accordance to the tweet. The word matrix is listed in Appendix B, Table B-2.

To obtain further insight on the TF-IDF word matrix and its relevance to each class label, the mean TF-IDF score is calculated.

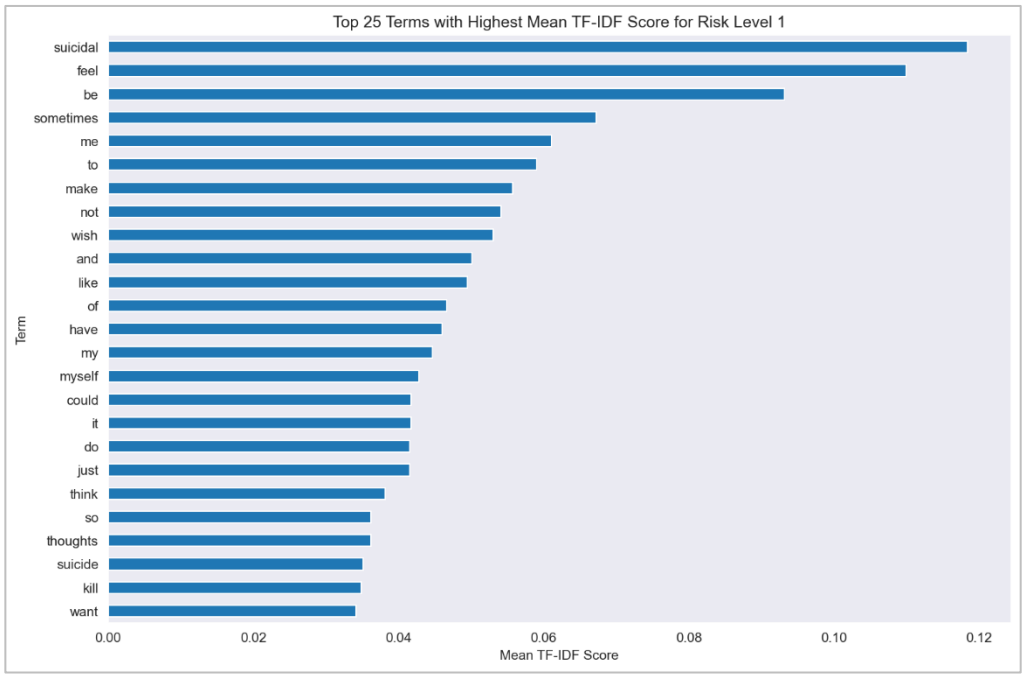


Figure 5.16: Top 25 Terms with Highest Mean TF-IDF score for Medium Suicide Risk

Figure 5.16 shows the 25 terms with the highest mean TF-IDF score for medium suicide risk level, with terms such as “suicidal”, “feel”, “make”, sometimes” and “thoughts” found to be dominant within the list. Some of these terms overlapped with the most commonly used terms within medium suicide risk, that was illustrated previously in Figure 5.3.

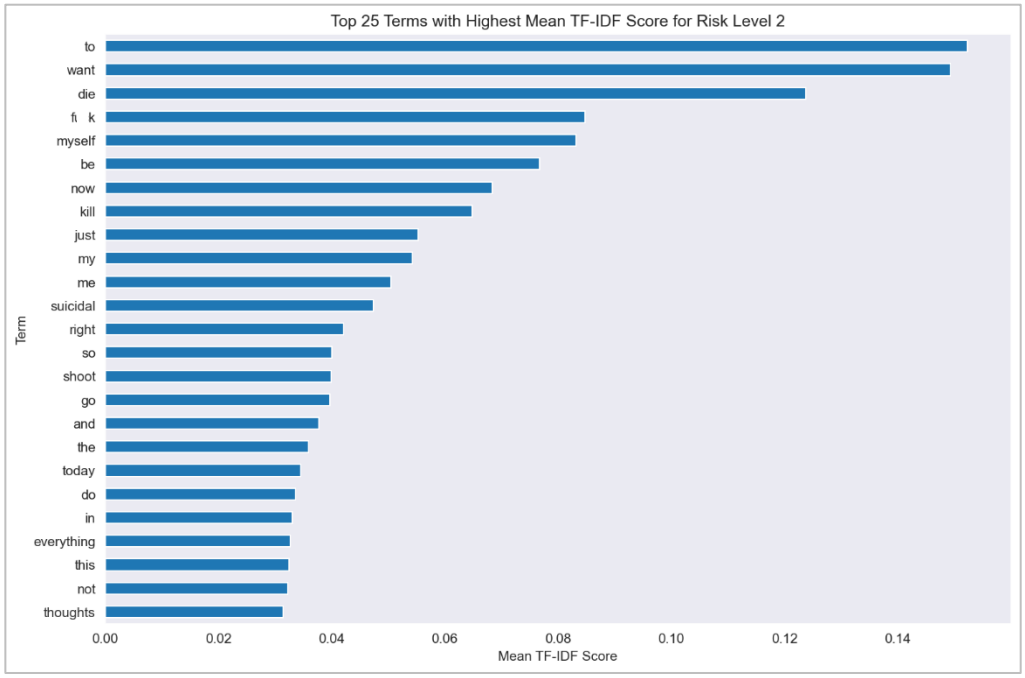


Figure 5.17: Top 25 Terms with Highest Mean TF-IDF Score for High Suicide Risk Level

For high suicide risk level, terms such as “want”, “die”, “now” and “kill” were found to be dominant within this list. Similarly, some overlap exists between these terms with the most commonly used terms shown previously in Figure 5.4.

5.4.3.1 Summary

As the TF-IDF algorithm measures the relevance of a given term by balancing the frequency of its occurrence within a single tweet with its rarity across the whole dataset. The resulting TF-IDF score reflects the importance of the term for a tweet in the dataset, while common and general stop words are given a low score. As a result, the terms shown in Figure 5.16 and Figure 5.17 have a high mean TF-IDF score since these terms appears frequently within an individual tweet, although it appears infrequently across the dataset.

However, some overlap still exists between the terms with the highest mean TF-IDF scores and the most frequently used terms for each level of suicide risk. For instance, for medium suicide risk, the terms "suicidal", "feel", and "thoughts" appears in highest mean TF-IDF scores, as does the most frequently used terms. For high suicide risk, the overlapping terms include

"want," "die," and "kill". This reveals that, alike the finding that was previously presented in Section 5.2, the terms considered to be important within each suicide risk level follows a distinct pattern, with terms in medium suicide risk inducing suicide ideation and terms in high suicide risk being more proactive in nature.

In addition to that, the TF-IDF score also revealed that most of the dominant terms among each suicide risk level were either noun or verbs. This is consistent with the findings on the grammatical properties, which was highlighted in Section 5.4.1. Interestingly, different adverbs of frequency and time were also prominently used among each suicide risk level, whereby "sometimes" was prominent within medium suicide risk level, while "right" and "now" were prominent within high suicide risk level.

5.5 Model Training

A train-test split of 80:20 split ratio was applied on the dataset, whereby former is used for training, while the hold out set is used for testing.

```
base_model = RandomForestClassifier(random_state = random.seed(1234))
base_model.fit(all_train_features,Y_train)
base_prediction=base_model.predict(all_test_features)
```

Figure 5.18: Model Training Code Snippet

Figure 5.18 shows the code snippet of the model training process. Firstly, the training set is fit into the Random Forest classification model. Thereafter, the model is checked against its prediction on the hold out test set. The evaluation and analysis of the model performance will be covered in Chapter 7.

CHAPTER 6

SYSTEM DEVELOPMENT AND MODEL INTEGRATION

6.1 Introduction

This chapter details the development of the web application that serves as an interface which facilitates the interaction between the integration of the APIs on the backend and its output on the frontend. Ultimately, the web application is a representation of existing social media sites to show how active monitoring of social media posts is achieved. The goal is to ensure that the suitable libraries and API endpoints are utilised and configured to work hand-in-hand to facilitate data transfer between the frontend and backend.

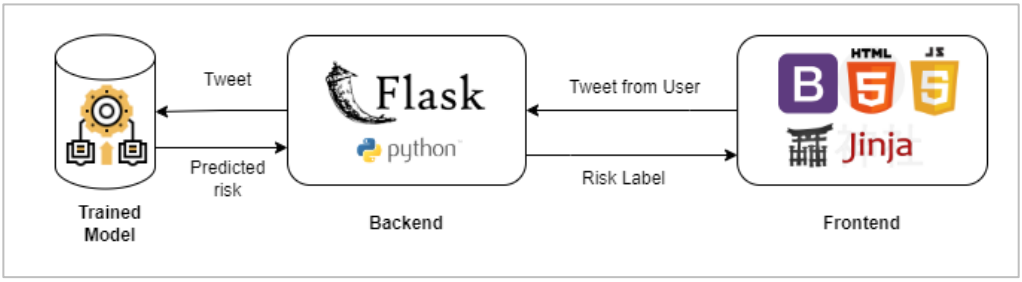


Figure 6.1: Overview of System Development

Based on Figure 6.1, the trained model interacts with the flask application on the backend Python environment, while the frontend utilises Bootstrap, HTML, JavaScript and Jinja2 templating engine to dynamically process the data received from the server and render it to the client during runtime.

6.2 Model Integration

In this section, the final, trained version of the model built in the previous chapter is integrated into the web application. To achieve that, the model needs to be deployed on python virtual environment in order to communicate with the web server. To get started, the pre-trained model is first exported as a pickle (.pkl) file. Pickle is a Python module that allows us to store the serialized object structure of the ML model. In this case, the pickle file

contains the pre-trained ML model which can be reloaded from python scripts outside of Jupyter Notebook. This is illustrated in the figure below.

```
import pickle
# save the model to disk
filename = 'model-final.pkl'
pickle.dump(model, open(filename, 'wb'))
```

Figure 6.2: Model Serialization Code Snippet

From there, the pre-trained model is loaded and deserialized by calling the *load* function from the *pickle* module. In the code snippet below, the pickle file which contains the pre-trained model is deserialized and assigned to the variable *model*.

```
model=pickle.load(open('model-final.pkl','rb'))
```

Figure 6.3: Model Deserialization Code Snippet

Once the pre-trained model is loaded, the prediction result is obtained by passing the the values of the feature set of the input tweet into the *model.predict()* function.

```
result = model.predict(all_features.values)
```

Figure 6.4: Model Prediction Code Snippet

6.3 System Development

This section outlines the details on the development of the complete system, which includes the logical handling of the data received from user input, as well as the display of information obtained from the backend server.

6.3.1 Backend Development

Flask is used to facilitate the communication between the client and server through HTTP requests. Table 6.1 illustrates the list of API endpoints defined.

Table 6.1: HTTP Endpoints used

HTTP Request Type	Route	Description
POST	/	• To obtain the prediction result based on the tweet • Calls functions to clean data, build and pass feature set into pre-trained model and refresh web application with predicted risk level
PUT	/<int:id>	• To set action according to user's selected response for high risk tweets • Handles the logic for storing the action associated with the tweet into the global data structure

6.3.1.1 POST Request

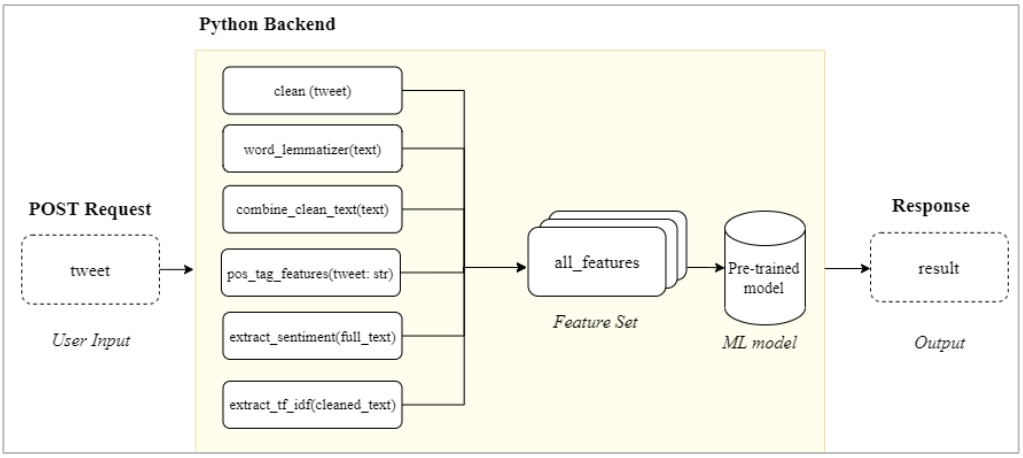


Figure 6.5: Overview on Backend Architecture for POST Request

On the client side, once a tweet is captured from the web application, the POST request is initiated, which calls the functions for data pre-processing, feature extraction and risk prediction. The pre-processing and feature extraction functions prepares and build the feature set that is passed to the pre-trained model. From there, the prediction result obtained from the pre-trained model is then passed back as an output reflected in the web application.

6.3.1.2 PUT Request

A PUT request is initiated once the moderator assigns an action to the tweet with high suicide risk. The backend server retrieves the data of the tweet ID and the selected action from the request header that is initiated through the PUT request on the frontend. From there, the action that corresponds to the tweet ID from the request is updated into the data structure. Then the system displays a feedback modal according to the data rendered to the template.

6.3.2 Frontend Development

6.3.2.1 View Dashboard

Figure 6.6 depicts the dashboard of the web application. Jinja2 template engine and Bootstrap were used to develop an intuitive and interactive interface that is dynamically updated in real-time.

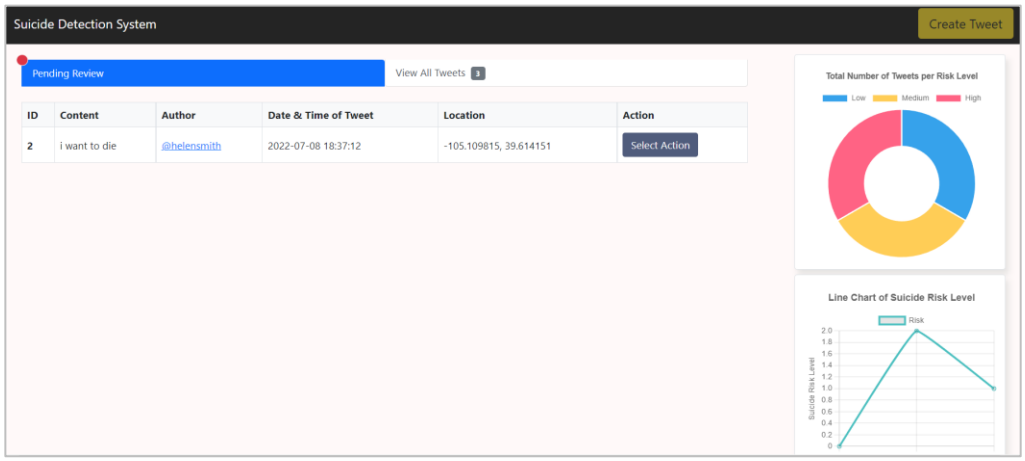


Figure 6.6: View Dashboard

Once the application is initialised on the server, the 2 main tab panes of “Pending Review” and “View All Tweets” are displayed. By default, the “Pending Review” tab is selected, which displays a table that consists of a detailed list of tweets that were labelled as high suicide risk level by the model. A red indicator will be shown whenever there are tweets that are flagged under the “Pending Review” tab.

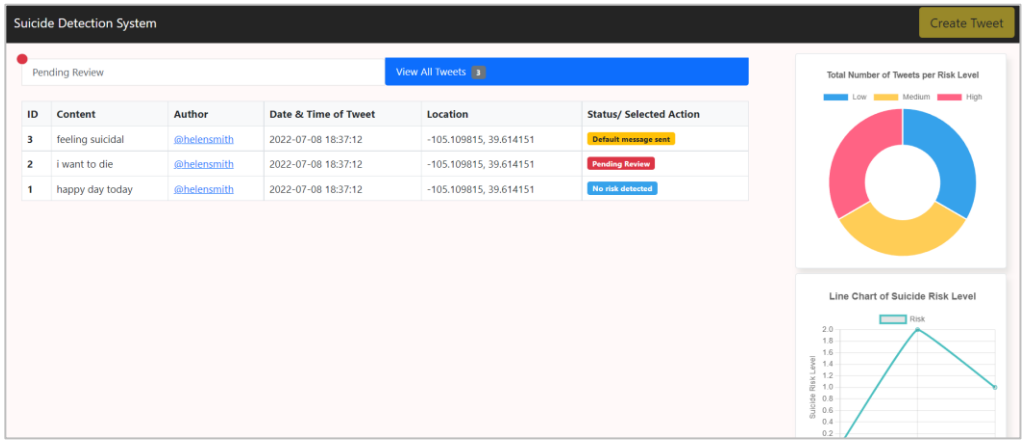


Figure 6.7: View Dashboard (View All Tweets Tab)

On the other hand, Figure 6.7 shows the "View All Tweets" tab which displays all the tweets that were detected by the system. The tweets were displayed with information on the content, author, date and time of tweet, location and its latest status/ selected action. In general, the colours for the status labels were customized according to each suicide risk whereby blue is used for low suicide risk; yellow for medium suicide risk and tweets with default message sent; red for high suicide risk and tweets escalated to the authorities.

6.3.2.2 View Real-time Statistics

Figure 6.8 depicts the real-time statistics visualized in a doughnut chart and line graph that were built using the Chart.js library.

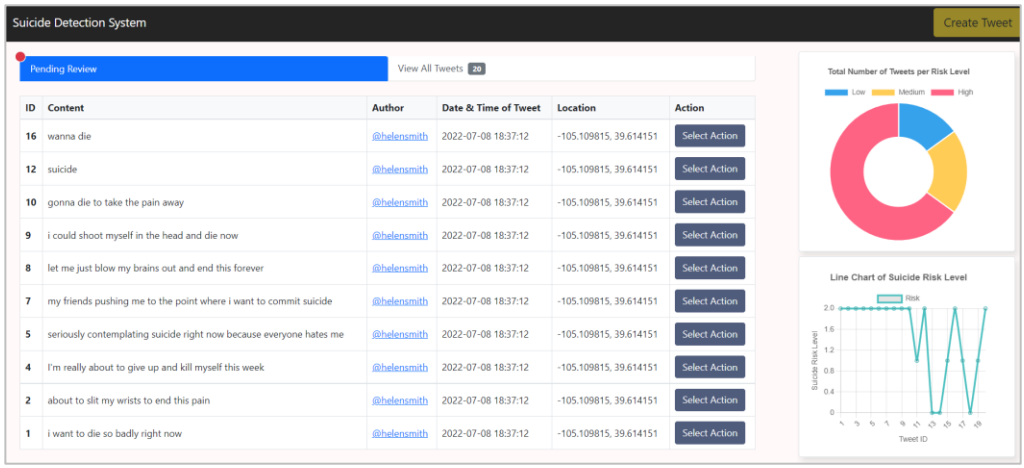


Figure 6.8: View Real-Time Statistics

The doughnut chart shows the breakdown of the total number of tweets according to its risk level, while the line graph illustrates the trend observed from the tweets over time.

6.3.2.3 Select Action

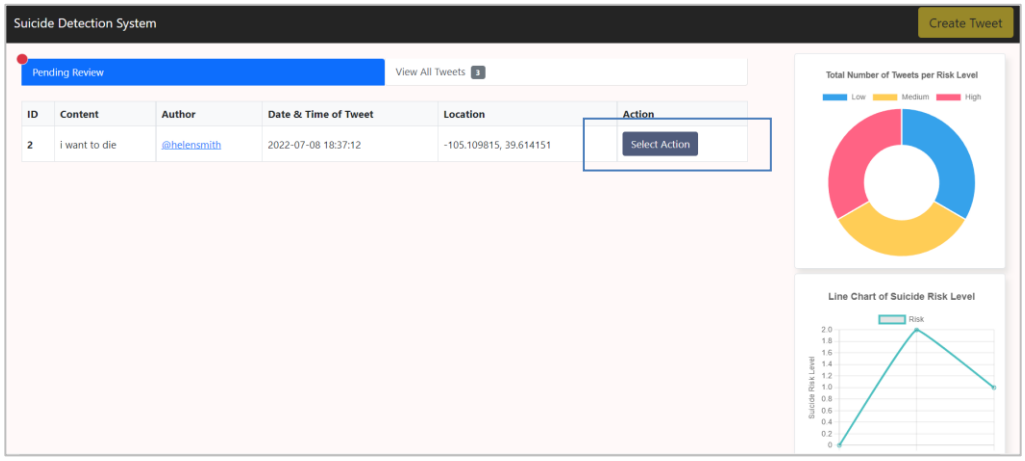


Figure 6.9: Select Action Button

Figure 6.9 shows the “Select Action” button for tweets flagged for review. From there, the moderator may click on the “Select Action” button to review the tweet content and select the appropriate action for the tweet. Once the tweet has been reviewed and assigned an action by the moderator, the tweet will be removed from the “Pending Review” tab.

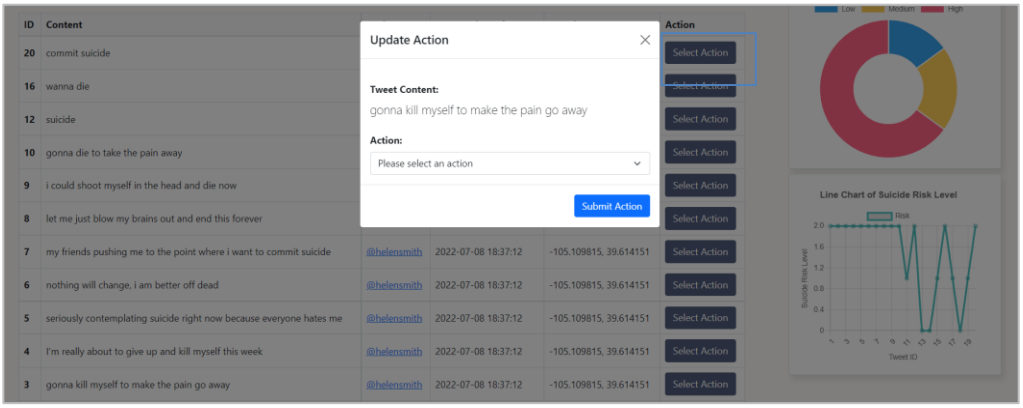


Figure 6.10: Select Action Modal

Once the moderator clicks on the “Select Action” button, a modal will be shown for them to assign an action for the tweet, as illustrated in Figure 6.10.

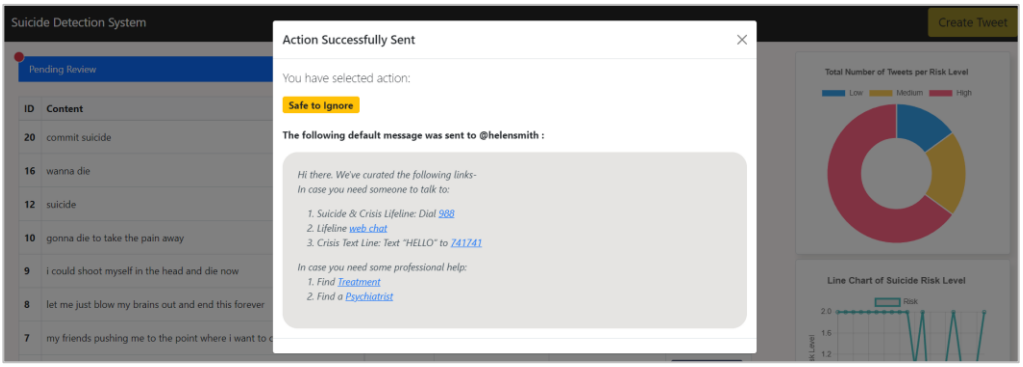


Figure 6.11: Response for Medium Suicide Risk

Figure 6.11 shows the response when “Safe to Ignore” is selected, where it shows that a default message was sent to the target user. The detailed content of the default message is also shown.

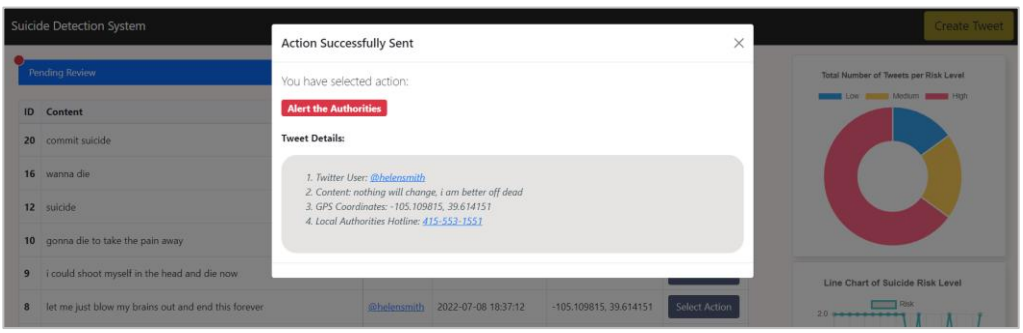


Figure 6.12: Response for High Suicide Risk

Figure 6.12 shows the response when “Alert the Authorities” is selected, where the details associated with the tweet such as the twitter username, tweet content, GPS coordinates and local authorities’ hotline were displayed. This is to supply relevant information to the moderator when they escalate this to the higher authority.

6.3.2.4 Update Action

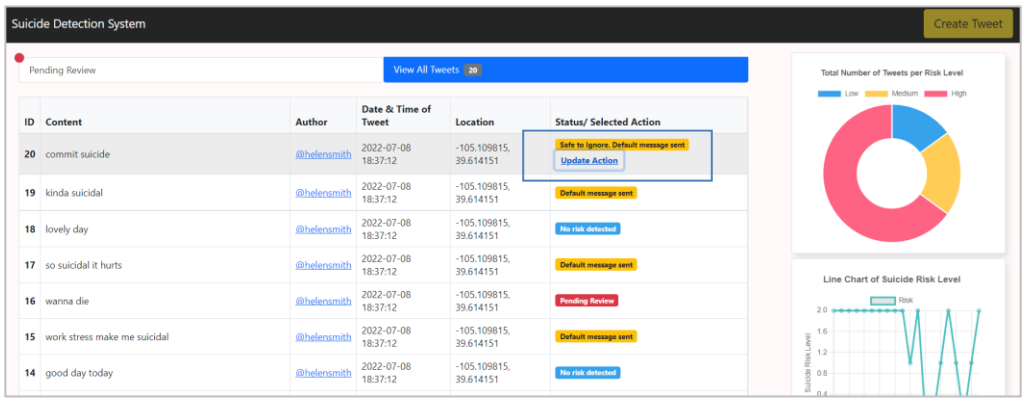


Figure 6.13: Update Action

If the moderator wishes to change the action that is associated with tweets with high suicide risk, they may click on the “Update Action” button, as illustrated in Figure 6.13.

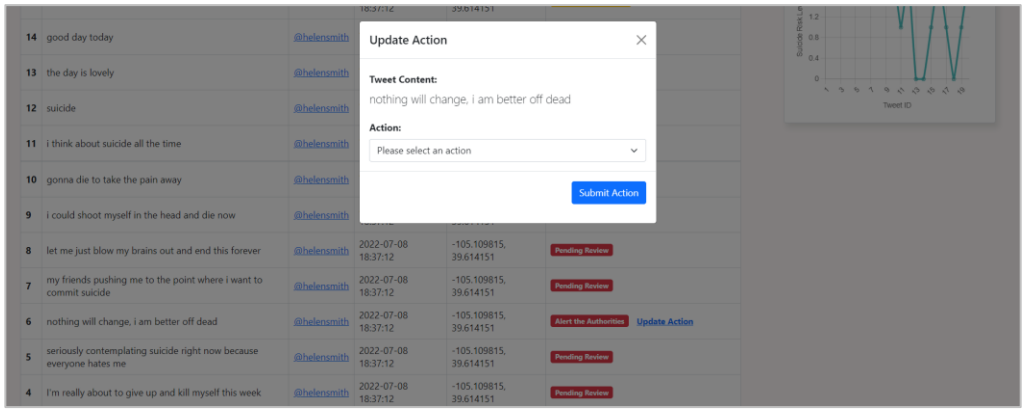


Figure 6.14: Update Action Modal

Figure 6.14 depicts the Update Action Modal that is shown when the "Update Action" button is clicked. The selected action from the drop-down list leads to the same response modal, as shown in section 6.3.2.3.

6.3.2.5 Create Tweet

Figure 6.15 shows the Create Tweet function that was created for the purpose of demonstrating the real-time tweets captured by the system. Once the tweet is created, this will trigger a POST request that sends the tweet to the Python backend server for processing.

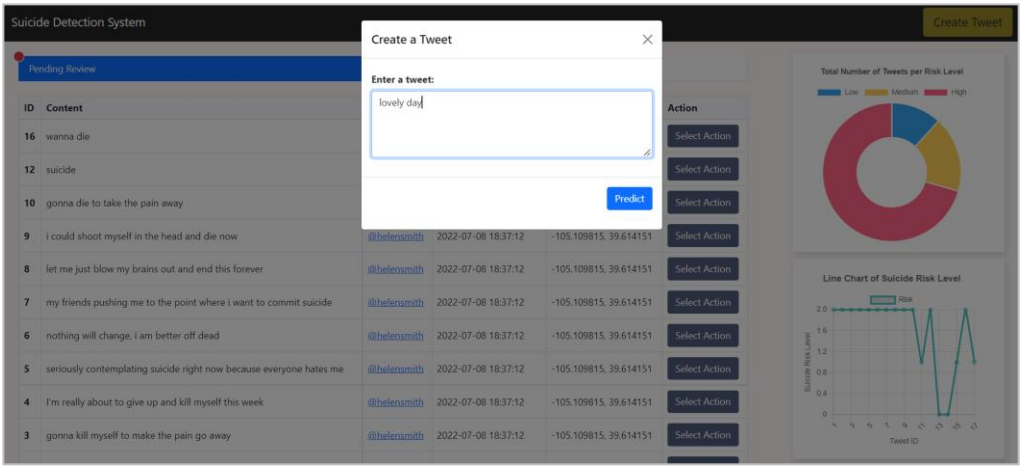


Figure 6.15: Create Tweet

To further enhance the demonstration of the output obtained from the backend server, the prediction result will be displayed to the user via a feedback modal. This is to provide visibility over the suicide risk level that was fetched from the server. The content within the modal is customized according to each risk level.

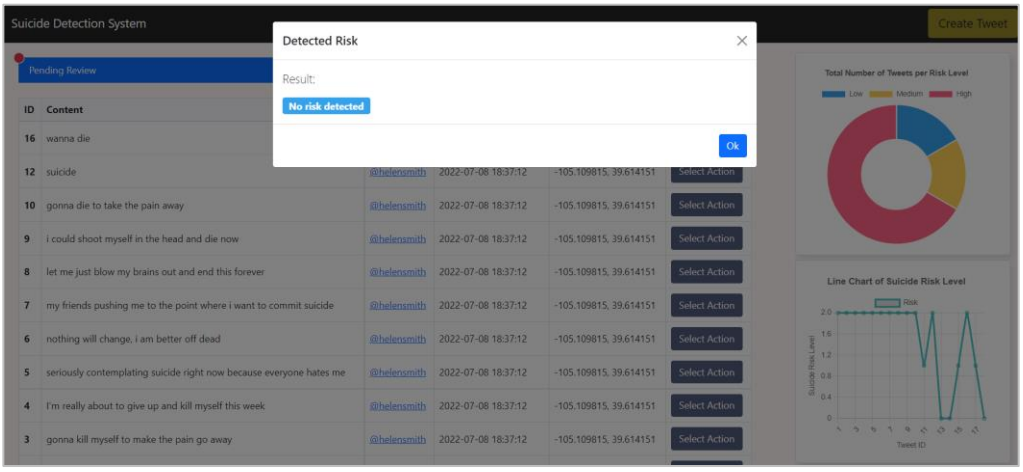


Figure 6.16: Feedback Modal for Low Suicide Risk

Figure 6.16 shows the feedback modal for tweets labelled with low suicide risk level. A blue “No risk detected” label is displayed, with no further actions or information.

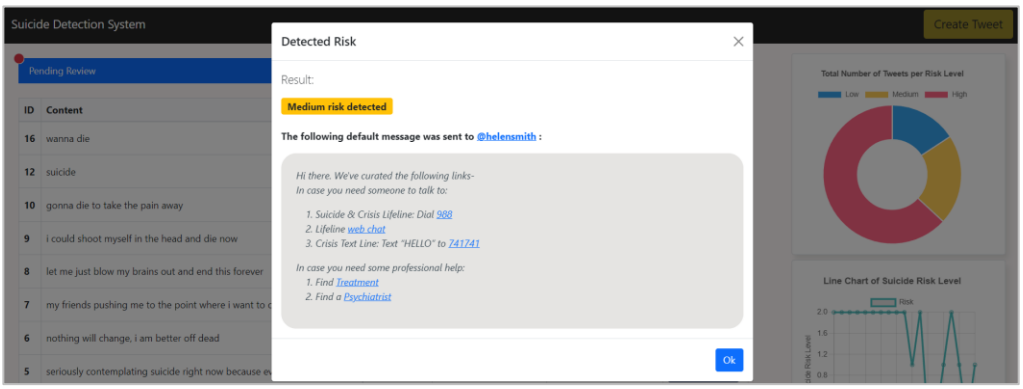


Figure 6.17: Feedback Modal for Medium Suicide Risk

Figure 6.17 shows the feedback modal for tweets labelled with medium suicide risk level A yellow “Medium risk detected” label is displayed, alongside with the detailed content of the default message that is sent to the target user. The mental health resources that were sent contains clickable redirect links to provide easy access for the user to visit these sites.

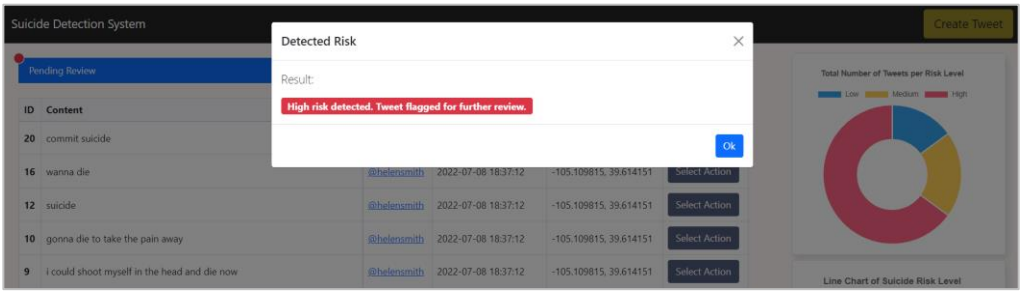


Figure 6.18: Feedback Modal for High Suicide Risk

Figure 6.18 depicts the feedback modal for tweets labelled with high suicide risk level. A red label that reads “High risk detected. Tweet flagged for further review” will be displayed.

6.4 Summary

In summary, various tools, modules and libraries were utilised during system development to streamline the data transfer between the backend server and frontend output. The pre-trained model was successfully deployed on the web server by using the *pickle* module which facilitates the serialization and deserialization of the pre-trained model so that it can be used in the web server. From there, API endpoints were setup to allow transfer of data between the client and the server. It was also validated that the web application was able to successfully utilise the pre-trained model to process and produce predictive results based on real-time data. From there, it was shown that the appropriate distress response was triggered successfully upon detection of tweets with medium to high suicide risk.

CHAPTER 7

RESULTS AND DISCUSSIONS

7.1 Introduction

This section presents the results, findings and its discussion into 2 main sub-sections. Firstly, the results obtained from the test set during the data modelling phase was analysed and benchmarked against the existing works. This is to assess and provide an understanding on the performance of the project's approach. Next, to evaluate and validate the efficacy of the system in real-time environment, system test was conducted using random test samples. The findings gathered from the system test were analysed to assess the performance of the system when random, unseen test samples are passed into the system. The performance results are evaluated based on the performance metrics of accuracy, precision, recall and discrepancy.

7.2 Model Performance Evaluation

In this section, the performance of the Random Forest model is evaluated based on its prediction on the hold out test set that consists of 138 samples. Comparative analysis was then conducted to understand the performance of the model compared to existing works.

7.2.1 Performance Analysis

The Random Forest model performed significantly well on the test set, yielding accuracy, precision, recall and discrepancy of 86.23%, 86.71%, 86.23% and 13.77% respectively.

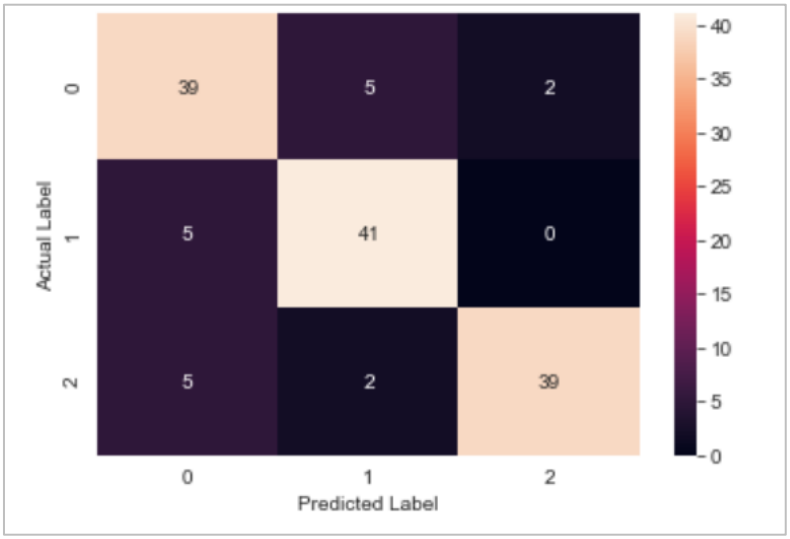


Figure 7.1: Confusion Matrix

Figure 7.1 shows the confusion matrix plotted to gain further insight on the model's performance. It was found that the true positives of medium suicide risk were the highest, which is 2 samples more than low and high suicide risk. It was also observed that the false positives were significantly higher across low and high suicide risk levels, such that the model frequently predicted tweets with low suicide risk to contain medium to high levels of suicide risk and tweets with high suicide risk to contain low and medium levels of suicide risk. In terms of false negatives, higher misclassification was found within the low and medium suicide risk level.

From these results, it is clear that the model's accuracy is affected by the feature set used by the model during its classification task, specifically the VADER sentiment score. To illustrate, a histogram of the output sentiment score from VADER is plotted according to its respective risk label, which is showed in Figure 7.2.

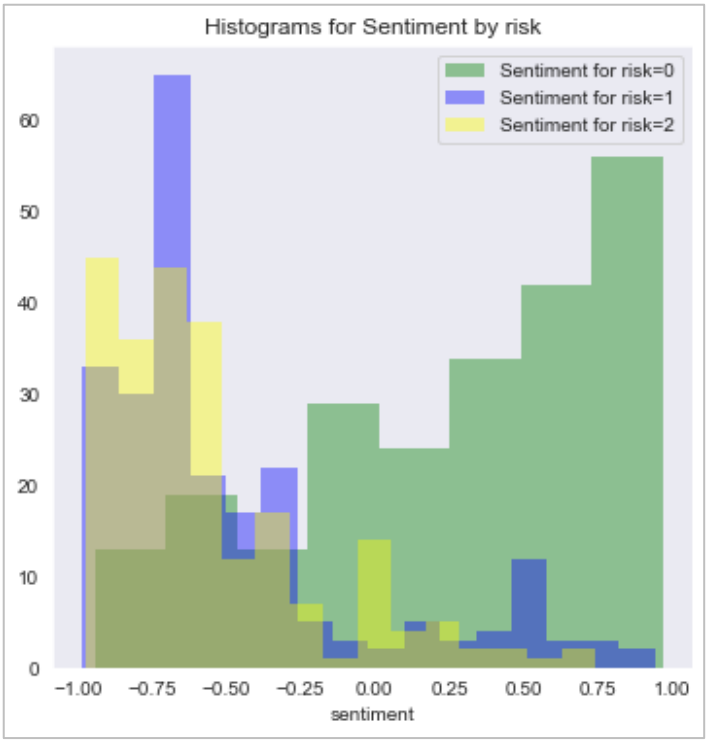


Figure 7.2: Histograms of Sentiment Score by Suicide Risk Level

Based on Figure 7.2, it is observed that the sentiment score for each risk level is distributed across the spectrum. In other words, there is no definitive boundary between each suicide risk level as the sentiment score from VADER varies across all levels. This finding is supported by the observation from section 5.4.2 which found that contradiction exists in the sentiment score from VADER, where the sentiment score does not always accurately mirror its associated suicide risk.

Despite that, there is a significant pattern observed, whereby samples with low suicide risk level are more likely to have a sentiment score more than -0.75, while samples with high suicide risk level are inclined to have a sentiment score that is less than 0.75. This finding shows that the model's high accuracy is attributed to its ability to leverage this pattern to classify tweets that are within the low and high suicide risk level. When considered from a reversed perspective, this pattern would also cause an adverse effect on the model performance. Such that, the outliers would result in the misclassification of samples, particularly in tweets of low and medium suicide risk.

To assess and gain an in-depth understanding on the performance of the Random Forest model, comparative analysis was carried out to evaluate the performance results of the model with the existing works. The existing works used for this analysis were extracted on the basis that these works also used Random Forest classifier in their approach. For better comparison, the analysis is divided into two sections: the first analyses the model's performance to works with multiclass classification while the second in binary classification. To better reflect the results obtained in the existing works, the performance metrics of some works were represented in range.

Table 7.1: Performance Results from Existing Works (Multiclass Classification)

Paper	**Data Source; Size of Dataset**	**Accuracy**	**Precision**	**Recall**
(Mbarek et al., 2019)	Twitter; 785 posts	-	81%	90%
(Nobles at al., 2018)	Text messages, based on voluntary participation; N/A	70%	81%	-
(Tadesse et al., 2019)	Reddit; 7201 posts	77.2% - 85.6%	76.3% - 85%	75.1% - 84%
(Muhammad Shah et al., 2020)	Reddit; 7098 posts	61.2% - 64.2%	60.6%- 62.7%	62.5% - 70.1%

From Table 7.1, it is observed that the model outperformed the others in terms of precision that ranges between 60.6% and 85%. In terms of accuracy, the model performed significantly better than the existing works with results between the range of 61.2% and 85.6%. On the other hand, the recall value was above average for the range of 62.5% - 90%.

Table 7.2: Performance Results from Existing Works (Binary Classification)

Paper	Data Source; Size of Dataset	Accuracy	Precision	Recall
(O'Dea et al., 2015)	Twitter; 1820 posts	-	77	65.67
(Hassan et al., 2020)	Conversational speech captured by Google Home Mini; N/A	76	77.12	91.67
(Shing et al., 2018)	Reddit; 865 user's post	-	67% - 70%	66% - 68%

For binary classification, the model performance was significantly higher than existing works by more than 10% across all 3 performance metrics which had an average of 76%, 74.21%, 74.78% respectively for accuracy, precision and recall. Hence, it is conclusively found that the model has a significantly higher performance results when compared with the existing works. In alignment with that, these findings have validated that the existing approach used have improved the classification task of detecting different levels of suicidality.

7.2.2 Summary

Based on the findings presented above, it is observed that the model yielded high performance results when evaluated with the performance metrics of accuracy, precision and recall. Further analysis revealed that there was a significant pattern observed from the sentiment score of the tweet samples, which has an effect on the model's performance. Through comparative analysis, it was found that the model performed significantly better than the existing works. Therefore, it is conclusively proven that the approach used to build the Random Forest model was effective such that it has improved the classification performance of detecting suicide ideation.

7.3 System Testing

This section details the findings obtained from the system test that was carried out to evaluate the systems performance in real-time.

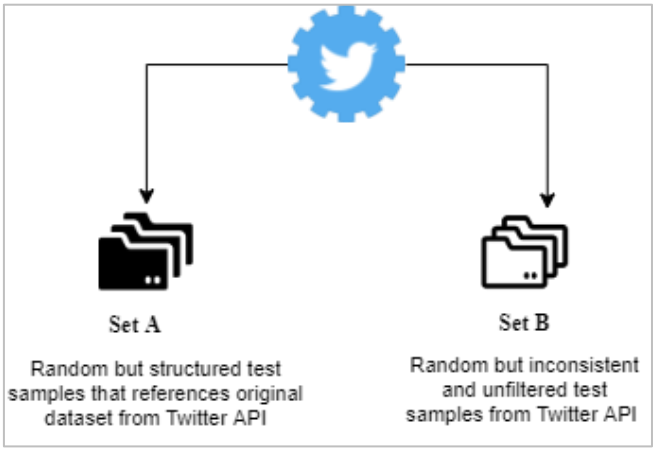


Figure 7.3: Overview on Test Samples Used

To get better insights on the system performance, the system was tested in real-time based on 2 datasets which is denoted as Set A and Set B. Set A consists of random test samples with references to the collected dataset while Set B consists of random test samples retrieved from random user input. These 2 sets of test samples are similar in the sense that it was drawn randomly from Twitter's API but differs slightly in the nature of its context.

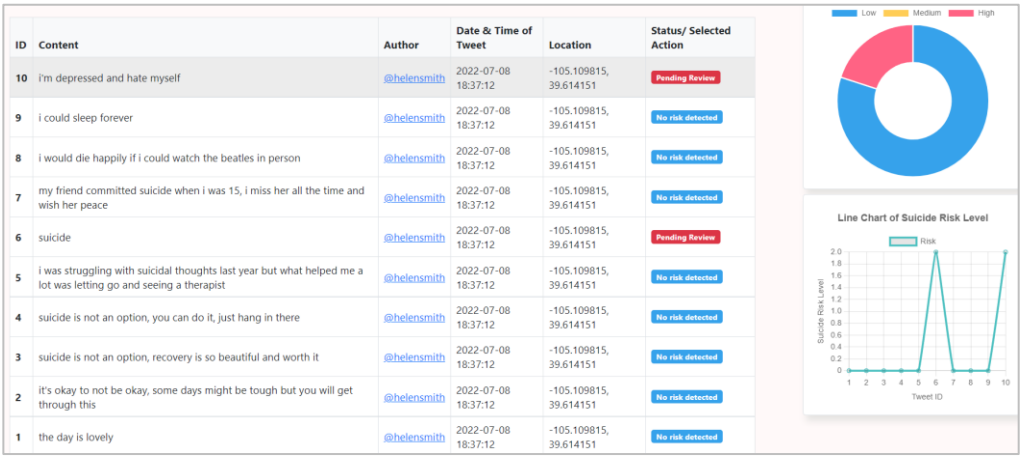
As the test samples from Set A references the collected dataset, these samples embody similar characteristics of the context used for model training. Furthermore, it was highlighted during the data collection phase, that this training dataset was further refined by removing redundant samples to provide better quality dataset to aid the model training process. Thus, the test samples in Set A would contain some level of consistency in its underlying context despite being unseen to the model.

On the other hand, Set B was obtained from random user input without any prior refinement, which implicitly represents inconsistent and unfiltered data. The main aim of conducting system test based on these 2 types of test samples is to obtain an unbiased validation and evaluation of the system's performance in real-time. Additionally, this would also allow us to understand the model's behaviour on unfiltered test samples. The actual test consists of more samples than what is shown in the following sub-sections.

7.3.1 Set A

7.3.1.1 Low Suicide Risk

Figure 7.4 shows the results obtained from the system for test samples of low suicide risk.



ID	Content	Author	Date & Time of Tweet	Location	Status/ Selected Action
10	i'm depressed and hate myself	@helensmith	2022-07-08 18:37:12	-105.109815, 39.614151	Pending Review
9	i could sleep forever	@helensmith	2022-07-08 18:37:12	-105.109815, 39.614151	No risk detected
8	i would die happily if i could watch the beatles in person	@helensmith	2022-07-08 18:37:12	-105.109815, 39.614151	No risk detected
7	my friend committed suicide when i was 15, i miss her all the time and wish her peace	@helensmith	2022-07-08 18:37:12	-105.109815, 39.614151	No risk detected
6	suicide	@helensmith	2022-07-08 18:37:12	-105.109815, 39.614151	Pending Review
5	i was struggling with suicidal thoughts last year but what helped me a lot was letting go and seeing a therapist	@helensmith	2022-07-08 18:37:12	-105.109815, 39.614151	No risk detected
4	suicide is not an option, you can do it, just hang in there	@helensmith	2022-07-08 18:37:12	-105.109815, 39.614151	No risk detected
3	suicide is not an option, recovery is so beautiful and worth it	@helensmith	2022-07-08 18:37:12	-105.109815, 39.614151	No risk detected
2	it's okay to not be okay, some days might be tough but you will get through this	@helensmith	2022-07-08 18:37:12	-105.109815, 39.614151	No risk detected
1	the day is lovely	@helensmith	2022-07-08 18:37:12	-105.109815, 39.614151	No risk detected

Figure 7.4: Test Samples for Low Suicide Risk

Based on the figure above, it is observed that the system is able to correctly classify 80% of the samples. This includes tweet that recalls their past suicide experiences, makes indirect references to suicide and mental health as well as the use of suicide indicative terms such as “die” in non suicide indicative context. In the 2 cases where the tweets were incorrectly labelled, the tweet either consisted of only one suicide indicative keyword or contained a higher number of negative terms despite its short tweet length. This finding suggests that the model is more effective in its classification task when more context is provided within the tweet. To test this hypothesis, Figure 7.5 shows a sample that contains a combination of terms with references to suicide indicative keywords such as “suicidal ideation” and “kill myself” as well as terms with positive connotations such as “enjoy” and “glad”. The predicted result shows that the tweet is of low suicide risk, which is correct.

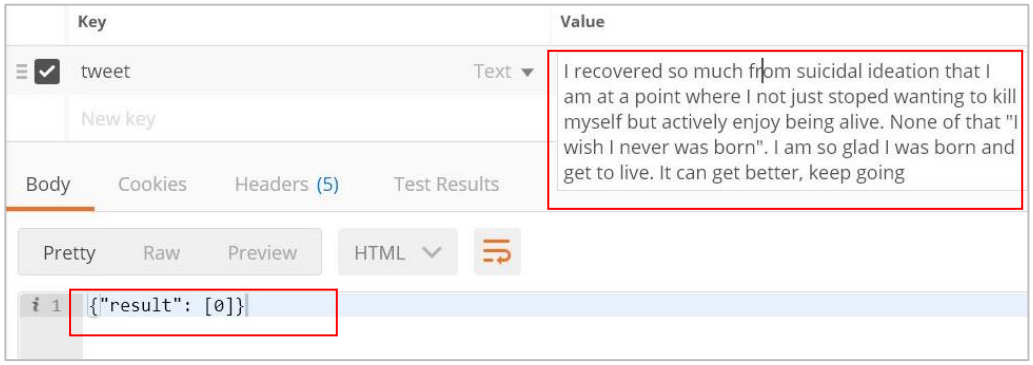


Figure 7.5: Additional Sample

As a result of these findings, the model appears to be more accurate when additional context is provided, as this gives the model greater flexibility in evaluating and classifying the suicide risk level.

7.3.1.2 Medium Suicide Risk

Figure 7.6 shows the detailed listing of the results obtained from the system for test samples of medium suicide risk.

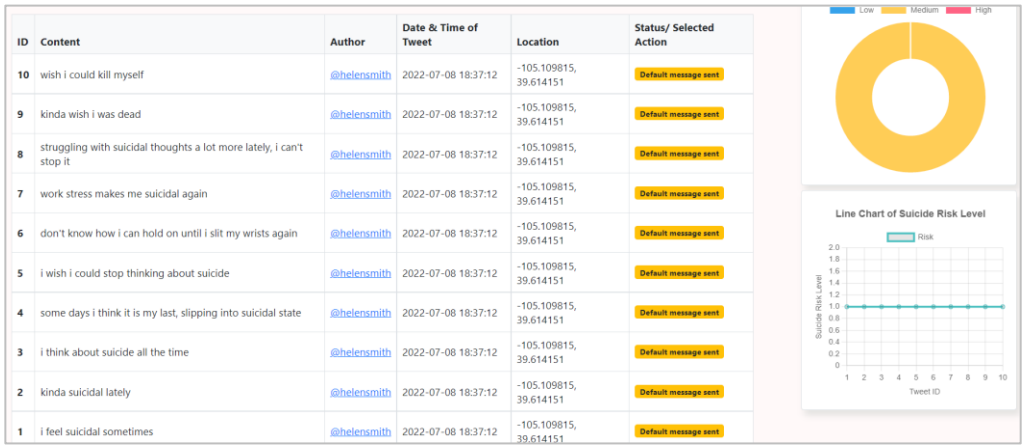


Figure 7.6: Test Samples for Medium Suicide Risk

In general, each sample includes different combination of terms such as “suicidal”, “feel” and “wish” that are adapted from tweet samples of medium suicide risk level. From Figure 7.6, it is gathered that the system is able to accurately classify 100% of the test samples.

7.3.1.3 High Suicide Risk

Figure 7.7 shows the detailed listing of the results obtained from the system for test samples of high suicide risk.

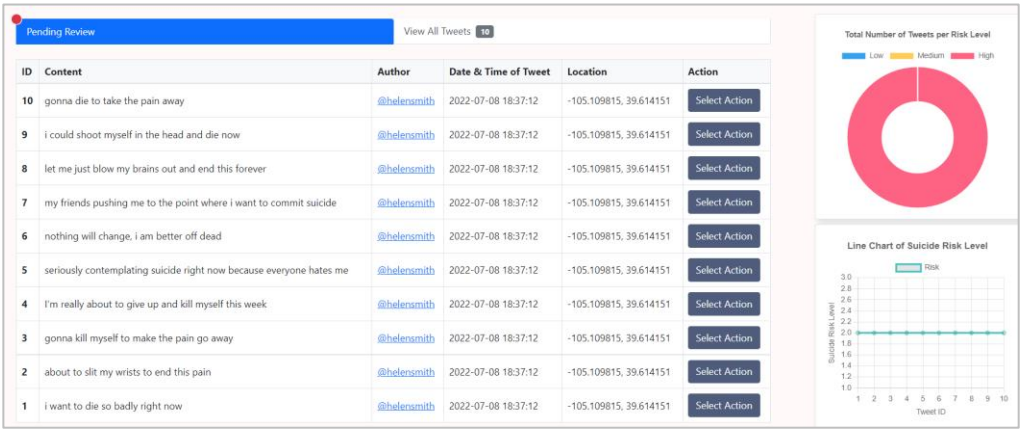


Figure 7.7: Test Samples for High Suicide Risk

In general, each sample includes different combination of terms such as “suicide”, “shoot myself” and “kill myself” that are adapted from tweet samples of high suicide risk level. As seen in Figure 7.7, all of the tweet samples were accurately classified according to its suicide risk level.

7.3.1.4 Summary

In summary, when validated against test samples that references the collected dataset, the model is able to effectively analyse the tweet and classify it to its corresponding suicide risk category, with accuracy, recall, precision and discrepancy of 97.77%, 97.77%, 97.92% and 2.23% respectively. The confusion matrix of the model performance for set A is shown in Figure 7.8.

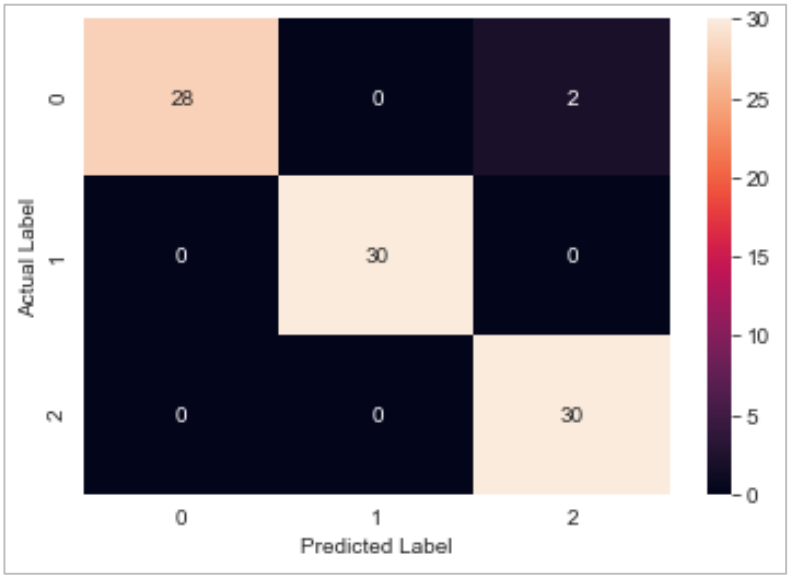


Figure 7.8: Confusion Matrix for Set A

The high accuracy in classifying tweets with medium and high suicide risk is attributed to the TF-IDF feature sets. While Set A consists of unseen test samples, the samples used for medium and high suicide risk level overlapped with the set of terms that were considered important for its suicide risk level. Interestingly, these samples were able to produce high performance results, which thus proves the ability of the model to utilise this information to analyse and identify terms that corresponds to a specific suicide risk category. As such, this demonstrates TF-IDF's ability to improve the quality of the feature set as it allows the model to leverage these scores to effectively distinguish between context of different suicide risk. In addition to that, the high accuracy also proves the model's ability to utilise the PoS tags feature set, particularly to recognize adverbs of frequency and time within tweets of medium and high suicide risk. However, the tweet should provide additional context in order for the model to be effective in analysing its context and relate the tweet to its appropriate suicide risk level. Otherwise, this would result in misclassification.

7.3.2 Set B

7.3.2.1 Performance Analysis

When system test was conducted based on test samples from Set B, which consists of data that is inconsistent in nature, it was found that the model was still capable of producing high performance results with accuracy, recall, precision and discrepancy of 89.33%, 92.98%, 83.08% and 10.67% respectively.

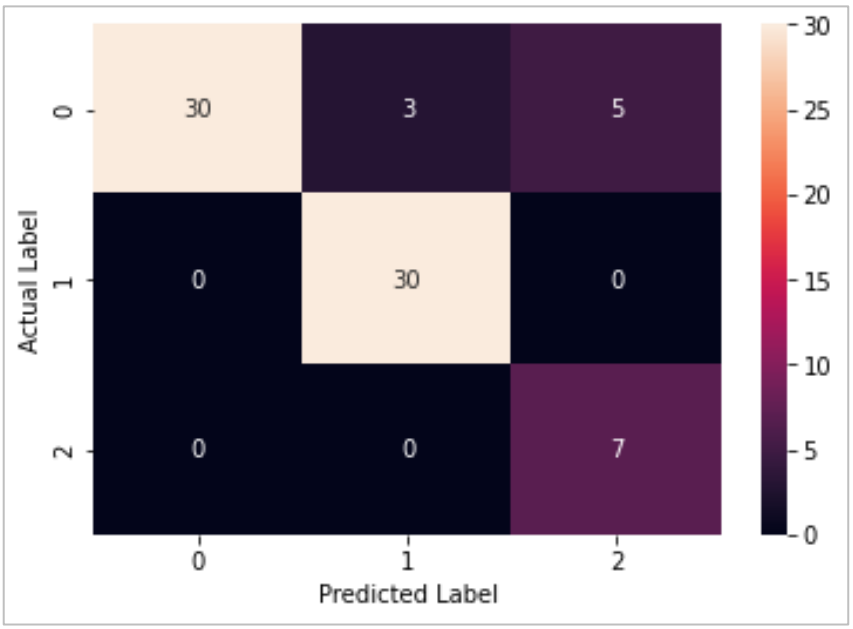


Figure 7.9: Confusion Matrix for Set B

Since the test samples are drawn from random and unfiltered user input, the general topic discussed in most of the tweets consists of random thoughts, replies and news articles. Hence, the distribution of the test samples is imbalanced, such that a majority of the test samples belongs to low and medium suicide risk level, with a minority of 7 test samples that belongs to high suicide risk level. From the confusion matrix shown in Figure 7.9, it is also observed that the model is able to accurately classify most tweets from each suicide risk level, such that the number of correctly classified samples greatly outweigh the number of misclassified samples of each class. In fact, misclassification only occurs at samples of low suicide risk level that are confused as medium and high suicide risk level by the model.

7.3.2.1.1 Correctly Classified Samples

@_heavenpostman Think about the love one in your life, the one that matters. Few years back had a suicidal thought. I was standing on the side of the road and about to jump in front of 2 speeding truck container.
@_heavenpostman Then a picture of my Mom came through, I step back and let my ticket to the after life passed by. Consider what's worth to die for in your life.

Figure 7.10: Samples of Correctly Classified Tweets

Figure 7.10 shows an excerpt of the test samples that were correctly classified by the model. In general, the authors of these tweets mentioned their past suicidal episodes. Although the tweet mentions different combination of suicide-indicative keywords, the model was still able to effectively analyse the tweet based on its entire context and classify its suicide risk level as low.

7.3.2.1.2 Incorrectly Classified Samples (Low as Medium)

An excerpt of test samples with low suicide risk that are misclassified by the model as medium suicide risk is shown in Figure 7.11.

thatâ€™s just some sh i cannot do
the media paints incels as malicious & egotistical instead of frustrated & suicidal
f my life

Figure 7.11: Samples of Misclassified Tweets

It is observed that the general context conveyed in the test samples are negative, whereby the first and last tweet uses profanities, while the second tweet consists of a combination of terms with negative connotation. As mentioned in Section 5.4.2, the sentiment score in VADER is affected by the language intensity used within the text. Given the strong language used by the tweet samples shown in the figure above, it is likely that these tweets would have high negative score computed from VADER. Furthermore, since the length of the tweet is short, there are no neutral or positive connotations to balance the context of the tweet sample, this would then result in a low compound score from VADER. Hence, the finding above proves that the misclassification is attributed to the sentiment score.

7.3.2.1.3 Incorrectly Classified Samples (Low as High)

@hellyjellybean @RahulKohli13 This is so f stupid lmao
@Hisoka_forever Damn that must be tough
@NenaThePothead mfs are sick i swear ðŸ¤¦â€â™€ï¸ðŸ˜,
im so tired
@stedecore_ This both healed and broke me at the same time and all I know is that Iâ€™m crying rn

Figure 7.12: Samples of Misclassified Tweets

Figure 7.12 shows an excerpt of test samples of low suicide risk that are misclassified by the model as high suicide risk. Similar to the misclassified test samples mentioned in the previous section (Section 7.3.2.1.2), these tweets either uses profanities or negative connotations. However, the misclassified tweets in Figure 7.12 contain more profanities than samples from Figure 7.11, which is the main difference between these samples. Hence, the increased

linguistic intensity results in a lower sentiment score, causing the model to misclassify these tweet samples to have a high suicide risk.

Besides that, the misclassification is also attributed to the TF-IDF scores. Since the TF-IDF features sets are based on the word matrix that the vectorizer was pre-trained with, it is limited to evaluate the tweet samples based on the terms that are found within the word matrix. Incidentally, while most of the terms used in the test samples above overlaps with the existing terms within the word matrix, these terms are mostly associated with negative emotions. Hence, the model misclassifies these samples as high suicide risk as the high TF-IDF score of these negative terms influences the inference of the model.

7.4 Summary

In summary, through direct comparison with existing works, it was found that the model performed significantly better in terms of accuracy, recall and precision. Besides that, it is widely known that performance evaluation is tricky in research works of any problem domain that involve data modelling process. This is because it is difficult for us to get an approximate idea on the model's behaviour in future settings based on the performance results obtained from the present test set. In recognition of this, this project has conducted system testing based on random test samples with varying context to balance the bias and obtain a fair and reliable evaluation on the system performance. In general, set A serves as a representation of the potential context that are used that translates to different levels of suicide risk, while set B represents a wider scope of social media data that is inconsistent in nature. Based on the findings, the model was still able to maintain its high performance in classifying the tweets into its respective suicide risk level at a low discrepancy. Hence, this further strengthens the basis that proves the efficacy of the developed system in real-time environment.

Upon further evaluation, it was proven that the application of NLP techniques such as PoS tagging, VADER sentiment analysis and TF-IDF produced high quality feature sets that improved the predictive ability of the model. Results from the analysis presented in this section showed that the

model was able to utilise the grammatical patterns found within the PoS tags to discern between tweets of medium and high suicide risk. Furthermore, it was evident that the model was able to leverage the pattern found within the VADER sentiment scores to accurately classify tweets of low and high medium risk. In particular, this includes accurately classifying low risk tweets that recall their past suicidal experiences and makes indirect references to suicide indicative keywords. At the same time, it was also found that the model was able to associate the patterns found within the TF-IDF score to discern key terms that are used to communicate different degrees of suicide ideation.

During system testing, it was found that the misclassification most commonly occurs within the false positives, whereby the tweets with low suicide risk are misclassified as either medium or high suicide risk. This is mainly attributed to the choice of words used to convey the context of the tweet, which has a direct influence on the sentiment score from VADER and TF-IDF feature sets. Hence, sufficient context is crucial for the model to perform effectively, as it provides more information for the model to learn from, thus resulting in more accurate suicide detection.

CHAPTER 8

CONCLUSIONS AND RECOMMENDATIONS

8.1 Conclusions

This project's main aim was to develop a comprehensive web application that serves as suicide detection and response system which caters to 3 different levels of suicide risk low, medium and high. For such purpose, this project covers the end-to-end activities from model development which uses Natural Language Processing and feature extraction techniques, to its deployment on Flask web application framework for real-time monitoring and detection of tweets, and finally the initiation of proactive responses tailored to specific suicide risk levels. Although previous works in the same problem domain have taken various approaches to address the challenge of detecting suicidal ideation, this project seeks to maximise the window of opportunity for early intervention by extending the system to include a proactive and tailored crisis intervention response mechanism that caters to specific suicide risk levels.

Through this project, the findings show that the approach used to train the Random Forest model outperformed existing works and was thus effective in discerning different levels of suicide risk. When tested with tweet samples of low, medium and high risk, it was proven that the system managed to proactively trigger the suitable distress response and flag tweets of high suicide risk for the further action. Most importantly, the system showed promising results when tested with random test samples, which proves the ability of the model generalize unseen data and thus verifies the efficacy of the system in real-time environment.

Through further analysis on the performance results, it was found that the model leveraged the patterns associated with PoS tags, VADER sentiment analysis and TF-IDF to discern between different levels of suicide risk, hence improving the predictive ability of the model. This includes correctly classifying low risk tweets that make indirect references to previous suicide attempts, as well as medium and high risk tweets that uses distinct key terms to communicate varying degrees of suicide ideation.

Analysis also showed that the discrepancy was mainly due to impact of the nature of the context such as the length of text and choice of words used on VADER's sentiment score and the TF-IDF score. These factors introduced contradiction and lead to ambiguity within the feature set which impacted the ability model to utilize this information for its classification task. Hence, it is better for more context to be provided as an input for the model so that the model can analyse and accurately classify the tweets into its respective suicide risk labels.

The limitation of this approach is that, since both the ML model and TF-IDF vectorizer was built from a pre-trained dataset, terms that are detected outside of the pre-trained datasets might have a negative impact on the performance of the model. This project attempted to address this gap by refining the training dataset used during model training to include data that covers a variety of suicide-indicative keywords. Despite that, this project recognizes the possibility of new terms that have yet to be trained on the ML model and TF-IDF vectorizer.

To shed light onto the issue of the expression of suicide ideation on social media, this project presented an approach that provides a basis for effective detection and response system that caters to different levels of suicide risk. Given the large volume of tweets that are posted daily, manual review of tweets is a challenging, time consuming and tedious task. Hence, the suicide detection and response system will be helpful towards early detection of such risks to bridge the gap between the individuals at risk and healthcare experts. Previous development of suicide detection systems was deemed controversial and heavily scrutinised by the general public and mass media for its rudimentary approach and ethical concerns surrounding personal data and privacy. While dealing with ethical challenges are always tricky, we should shift our emphasis towards recognizing the underlying value of taking a deliberate approach to address this issue. With that in mind, this project also acknowledges the fact that there is no definitive, one size fits all solution that completely addresses ethical issues. Ultimately, this project seeks to balance between the need to address the issue of suicide ideation expression on social

media and the long-term values associated with the sustainability of this suicide prevention tool.

With the best interest of the mental wellbeing of the online community as the guiding principle of this project, this ethical challenge was addressed from a constructive viewpoint by introducing an approach that is carefully formulated and fully transparent, leaving no doubts for explicit use or exploitation. It is with hopes that this system could contribute significant value that drives the community one step closer towards achieving the sustainable development goals of good mental health and well-being.

8.2 Recommendations for future work

For future recommendations, it is recommended for active learning machine learning approach to be leveraged to further improve the performance of the model. Manual data annotation requires a large amount of time and effort that eventually leads to a high cost. Hence, active learning can be utilised to optimise the learning process. This would allow the model to gradually learn more efficiently as time passes. Furthermore, as the current approach only caters to data from a specific region, it is recommended for the work to be tested for other regions to give it a better understanding of what to flag since the usage of terms, slangs and structure of the context might vary.

REFERENCES

Adenowo, A. and Adenowo, B., 2013. Software engineering methodologies: a review of the waterfall model and object- oriented approach. *International Journal of Scientific and Engineering Research*, 4(7), pp.427-434.

Ahuja, R., Chug, A., Kohli, S., Gupta, S., & Ahuja, P., 2019. The impact of features extraction on the sentiment analysis. *Procedia Computer Science*, 152, pp.341-348.

Bagwan, K. and Ghule, S., 2019. A modern review on laravel- php framework. *Iconic Research And Engineering Journals*, 2 (12), pp.1-3.

Boudreaux, E. D., Rundensteiner, E., Liu, F., Wang, B., Larkin, C., Agu, E., Ghosh, S., Semeter, J., Simon, G., & Davis-Martin, R. E., 2021. Applying machine learning approaches to suicide prediction using healthcare data: overview and future directions. *Frontiers in psychiatry*, 12, pp.1-7.

Calvo, R., Milne, D., Hussain, M. And Christensen, H., 2017. Natural language processing in mental health applications using non-clinical texts. *Natural Language Engineering. Cambridge University Press*, 23(5), pp. 649–685.

Chung, C. and Pennebaker, J., 2012. Linguistic inquiry and word count (liwc): pronounced “luke,” . . . And other useful facts. [e-book] Pennsylvania: IGI Global, Available at: <https://www.igi-global.com/gateway/chapter/61050> [Accessed 3 March 2022].

Coppersmith, G., Leary, R., Crutchley, P. and Fine, A., 2018. Natural language processing of social media as screening for suicide risk. *Biomedical Informatics Insights*10, pp.1-11.

Dunlop, S., More, E. and Romer, D., 2011. Where do youth learn about suicides on the Internet, and what influence does this have on suicidal ideation?. *Journal of Child Psychology and Psychiatry*, 52(10), pp.1073-1080.

Dineva, K. and Atanasova, T., 2020. Systematic look at machine learning algorithms advantages, disadvantages and practical applications. In: 20th International Multidisciplinary Scientific GeoConference SGEM 2020. Sofia, Bulgaria, 20 June 2020. Sofia: Surveying, Geology and Mining, Ecology and Management (SGEM).

Eklund, M., 2018. ‘Comparing feature extraction methods and effects of pre-processing methods for multi-label classification of textual data’. Masters thesis, KTH Royal Institute of Technology, Stockholm.

Fullerton, J., 2019. Teenage girl kills herself 'after Instagram poll' in Malaysia. [online] the Guardian. Available at:

<https://www.theguardian.com/world/2019/may/15/teenage-girl-kills-herself-after-instagram-poll-in-malaysia> [Accessed 9 March 2022].

Gomes de Andrade, N., Pawson, D., Muriello, D., Donahue, L. and Guadagno, J., 2018. Ethics and artificial intelligence: suicide prevention on Facebook. *Philosophy & Technology*, 31 pp. 669-684.

Hassan, S., Hassan, S. and Zakia, U., 2020. *Recognizing suicidal intent in depressed population using nlp: a pilot study*. In: 2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). Vancouver, British Columbia, Canada, 4-7 November 2020.

Hassonah, M., Al-Sayyed, R., Rodan, A., Al-Zoubi, A., Aljarah, I. and Faris, H., 2020. An efficient hybrid filter and evolutionary wrapper approach for sentiment analysis of various topics on Twitter. *Knowledge-Based Systems*, 192. Available through: UTAR Library website <https://library.utar.edu.my/> [Accessed 10 March 2022].

Hong, L., Convertino, G. and H. Chi, E., 2011. *Language matters in twitter: a large scale study*. In: Fifth International AAAI Conference on Weblogs and Social Media. Barcelona, Spain, 17-21 July 2011. California: Association for the Advancement of Artificial Intelligence.

Hutto, C., & Gilbert, E. (2014). VADER: a parsimonious rule-based model for sentiment analysis of social media text. In: Proceedings of the International AAAI Conference on Web and Social Media. Michigan, USA, 1-4 July 2014. California: Association for the Advancement of Artificial Intelligence.

Ji, S., Pan, S., Li, X., Cambria, E., Long, G. and Huang, Z., 2021. Suicidal ideation detection: a review of machine learning methods and applications. *IEEE Transactions on Computational Social Systems*, 8(1), pp.214-226.

Ji, S., Yu, C., Fung, S., Pan, S. and Long, G., 2018. Supervised learning for suicidal ideation detection in online user content. *Complexity* 2018, pp.1-10.

Kemp, S., 2021. *Digital 2021 October Global Statshot Report*. [online] DataReportal. Available through DataReportal: <https://datareportal.com/reports/digital-2021-october-global-statshot> [Accessed 23 February 2022].

Korani, W., & Mouhoub, M., 2022. *Sentiment analysis of serious suicide references in twitter social network.* In: 12th International Conference On Agents And Artificial Intelligence. Valletta, Malta, 22-24 February 2020. Portugal: SciTePress.

Kubben, P., Dumontier, M. and Dekker, A., 2018. Fundamentals of Clinical Data Science. [e-book] Cham: Springer. Available at: <https://library.oapen.org/handle/20.500.12657/22918> [Accessed 1 April 2022].

Liu, X., Liu, X., Sun, J., Yu, N., Sun, B., Li, Q. and Zhu, T., 2019. Proactive suicide prevention online (PSPO): machine identification and crisis management for chinese social media users with suicidal thoughts and behaviors. *Journal of Medical Internet Research*, 21(5), e.11705.

Luxton, D., June, J. and Fairall, J., 2012. Social media and suicide: a public health perspective. *American Journal of Public Health*, 102(S2), pp.S195-S200.

Maratkar, P. and Adkar, P., 2021. React JS – an emerging frontend javascript library. *Iconic Research And Engineering Journals*, 4(12), pp. 99-102.

Meyer, P., 2021. Natural Language Processing Tasks. [online] Medium. Available at: <https://towardsdatascience.com/natural-language-processing-tasks-3278907702f3> [Accessed 10 April 2022].

Maken, P., Gupta, A. and Gupta, M., 2019. A study on various techniques involved in gender prediction system: a comprehensive review. *Cybernetics and Information Technologies*, 19 (2), pp.51-73

Mbarek, A., Jamoussi, S., Charfi, A. and Ben Hamadou, A., 2019. *Suicidal profiles detection in Twitter*. In: Proceedings of the 15th International Conference on Web Information Systems and Technologies (WEBIST 2019), Vienna, Austria 18-20 September 2019. Portugal: SciTePress.

McClain, C., Widjaya, R., Rivero, G. and Smith, A., 2022. *The Behaviors and Attitudes of U.S. Adults on Twitter.* [online] Pew Research. Available through: Pew Research <https://www.pewresearch.org/internet/wp-content/uploads/sites/9/2021/11/PDL_11.15.21_Twitter_users_final_report.pdf> [Accessed 23 February 2022].

Millner, A., Lee, M. and Nock, M., 2016. Describing and measuring the pathway to suicide attempts: a preliminary study. *Suicide and Life-Threatening Behavior*, 47(3), pp.353-369.

Muhammad Shah, F., Haque, F., Un Nur, R., Al Jahan, S. and Mamud, Z., 2020. A hybridized feature extraction approach to suicidal ideation detection from social media post. *2020 IEEE Region 10 Symposium (TENSYMP)*, pp.985-988.

Nobles, A., Glenn, J., Kowsari, K., Teachman, B. and Barnes, L., 2018. *Identification of imminent suicide risk among young adults using text messages*. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. Montreal, Canada 21-27 April 2018. New York, Associate for Computing Machinery.

Nordin, N., Zainol, Z., Mohd Noor, M. and Chan, L.F., 2021. A comparative study of machine learning techniques for suicide attempts predictive model. *Health Informatics Journal*, 27(1).

O'Dea, B., Wan, S., Batterham, P., Calear, A., Paris, C. and Christensen, H., 2015. Detecting suicidality on Twitter. *Internet Interventions*, 2(2), pp.183-188.

Qudus Khan, F., Rasheed, S., Alsheshtawi, M., Mohamed Ahmed, T. and Jan, S., 2020. A comparative analysis of rad and agile technique for management of computing graduation projects. *Computers, Materials and Continua*, 64(2), pp.777-796.

Parrott, S., Britt, B., Hayes, J. and Albright, D., 2020. Social media and suicide: a validation of terms to help identify suicide-related social media posts. *Journal of Evidence-Based Social Work*, 17(5), pp.624-634.

Pinto, A., Oliveira, H., & Alves, A., 2016. Comparing the performance of different nlp toolkits in formal and social media text. *5th Symposium On Languages, Applications And Technologies*, 3, pp.3:1–3:16.

Pourmand A, Roberson J, Caggiula A, Monsalve N, Rahimi M, Torres-Llenza V., 2019. Social media and suicide: a review of technology-based epidemiology and risk assessment. *Telemedicine journal and e-health : the official journal of the American Telemedicine Association*, 25(10), pp.880–888.

Rabani, S., Khan, Q. and Khanday, A., 2020. Detection of suicidal ideation on twitter using machine learning & ensemble approaches. *Baghdad Science Journal*, 17(4), p.1328-1339.

Resnik, P., Foreman, A., Kuchuk, M., Musacchio Schafer, K. and Pinkham, B., 2020. Naturally occurring language as a source of evidence in suicide prevention. *Suicide and Life-Threatening Behavior*, 51(1), pp.88-96.

Roshen, G., 2017. Getting Our Community Help in Real Time | Meta. [online]. Available at: <https://about.fb.com/news/2017/11/getting-our-community-help-in-real-time/> [Accessed 3 March 2022].

Sadilek, A., Homan, C., Lasecki, W.S., Silenzio, V.M., & Kautz, H.A., 2013. Modeling Fine-Grained Dynamics of Mood at Scale.

Schlosser, S., Toninelli, D. and Cameletti, M., 2021. Comparing methods to collect and geolocate tweets in Great Britain. *Journal of Open Innovation: Technology, Market, and Complexity*, 7(1), p.44.

Shetty, J., Dash, D., Joish, A. and C, G., 2020. review paper on web frameworks, databases and web stacks. *International Research Journal of Engineering and Technology (IRJET)*, 7(4), pp.5734-5738.

Shing, H., Nair, S., Zirikly, A., Friedenberg, M., Daumé III, H. and Resnik, P., 2018. *Expert, Crowdsourced, and Machine Assessment of Suicide Risk via Online Postings*. In: Proceedings of the Fifth Workshop on Computational Linguistics and Clinical Psychology: From Keyboard to Clinic, New Orleans,

United States June 2018. New Orleans, LA. Association for Computational Linguistics.

Soron, T., 2019. “I will kill myself” – The series of posts in Facebook and unnoticed departure of a life. *Asian Journal of Psychiatry*, 44, pp.55-57.

Soron, T. and Shariful Islam, S., 2020. Suicide on Facebook-the tales of unnoticed departure in Bangladesh. *Global Mental Health*, 7, p. e12.

Surve, N., 2019. Sentiment analysis using Natural Language Processing (NLP). *International Research Journal of Engineering and Technology (IRJET)*, 6(9), pp.1240-1244.

Tadesse, M., Lin, H., Xu, B. and Yang, L., 2019. Detection of Suicide Ideation in Social Media Forums Using Deep Learning. *Algorithms*, 13(1), 7.

Tasse, D., Liu, Z., Sciuto, A. and Hong, J., 2017. *State of the Geotags: Motivations and Recent Changes*. In: Proceedings of the International AAAI Conference on Web and Social Media, Québec, Canada, 15-18 May 2017. California: The AAAI Press.

Van Casteren, W., 2017. The Waterfall Model and the agile Methodologies: A comparison by project characteristics. [online] Research Gate. Available at: < https://doi.org/10.13140/RG.2.2.36825.72805 > [Accessed 14 February 2022].

Vaughn, S., 2022. *Twitter’s self-harm flag feature is problematic*. [online]. Available at: <https://medium.com/we-need-to-talk/the-problematic-moderation-of-twitters-self-harm-flag-feature-df9002fbf5b8> [Accessed 14 February 2022].

Wankhade, M., Rao, A. and Kulkarni, C., 2022. A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review*, 2022.

Waykole, R. and Thakare, A., 2018. A review of feature extraction methods for text classification. *International Journal of Advance Engineering and Research Development*, 5(4), pp.351-354.

World Health Organization, 2018. Health workforce: fact sheet on Sustainable Development Goals (SDGs): health targets. [online] Available at: <https://apps.who.int/iris/handle/10665/340830> [Accessed 1 August 2022].

World Health Organization, 2021. *Suicide*. [online] Available at: <https://www.who.int/news-room/fact-sheets/detail/suicide> [Accessed 20 February 2022].

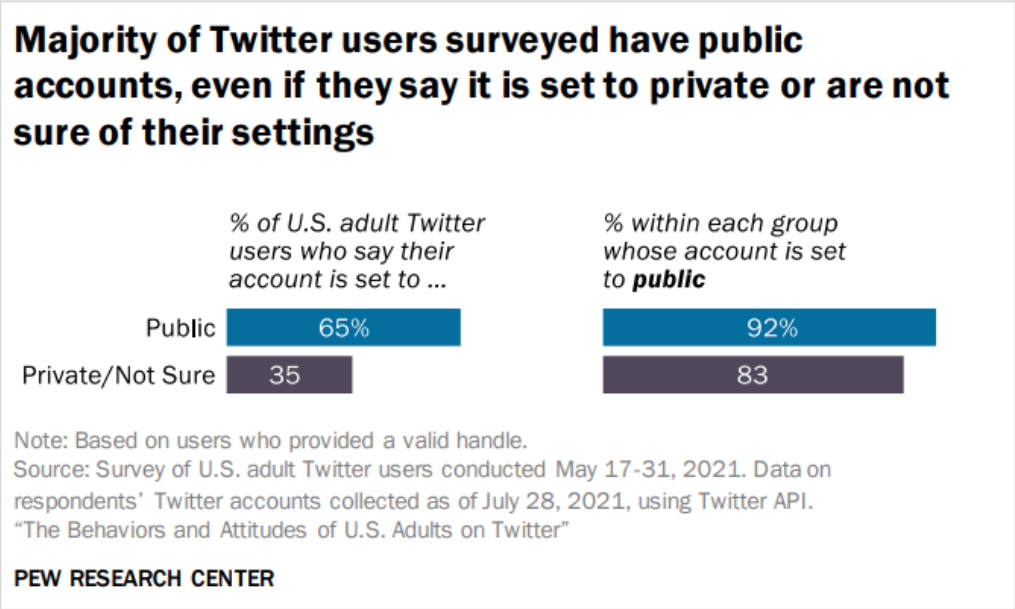
Yang, B., Xia, L., Liu, L., Nie, W., Liu, Q., Li, X., Ao, M., Wang, X., Xie, Y., Liu, Z., Huang, Y., Huang, Z., Gong, X. and Luo, D., 2021. A suicide

monitoring and crisis intervention strategy based on knowledge graph technology for “tree hole” microblog users in china. *Frontiers in Psychology*, 12.

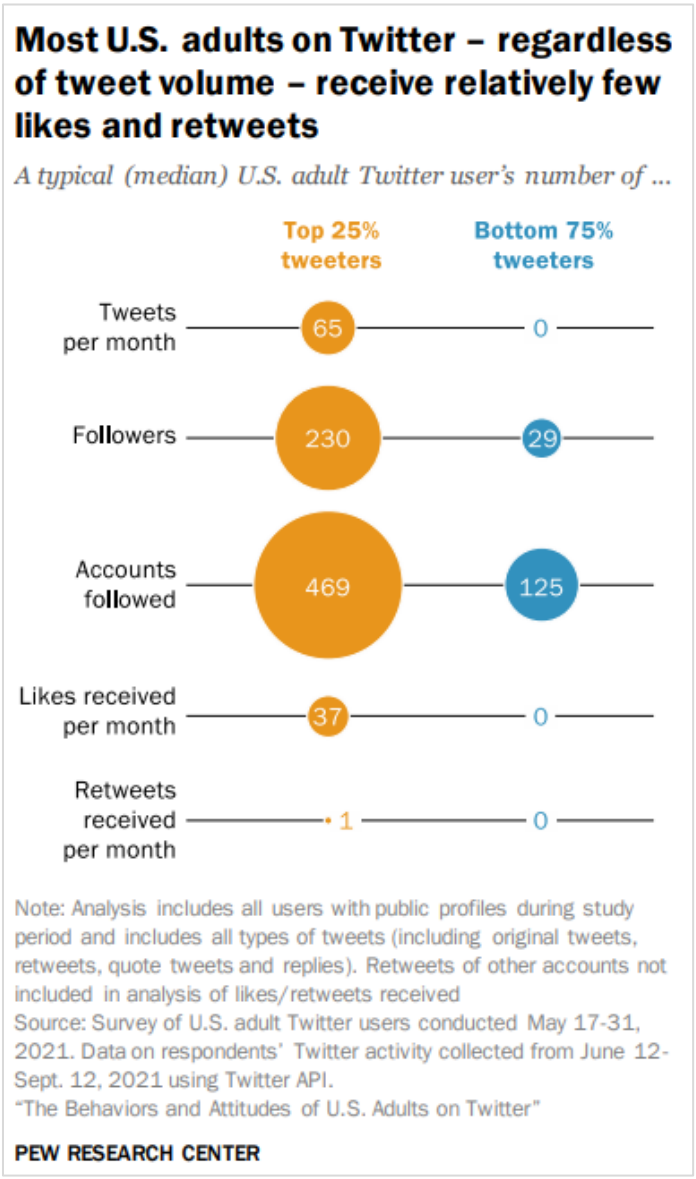
Zikopi, E., 2019, *A case study research on scrum framework*. Masters. KTH Royal Institute of Technology. Available at: <http://kth.diva-portal.org/smash/get/diva2:1337239/FULLTEXT01.pdf> [Accessed 15 April 2022].

APPENDICES

Appendix A: Graphs



Graph A-1: United States Twitter adult users account settings (McClain et al., 2022).



Graph A-2: United States Twitter adult users tweet volume per month (McClain et al., 2022).

Appendix B: Tables

Table B-1: List of PoS tags abbreviations

No.	PoS Tag	Description
1	CC	coordinating conjunction
2	CD	cardinal digit
3	DT	determiner
4	EX	existential
5	FW	foreign word
6	IN	preposition/subordinating conjunctio
7	JJ	adjective
8	JJR	adjective, comparative
9	JJS	adjective, superlative
10	LS	list marker
11	MD	modal
12	NN	Noun, singular
13	NNS	Noun plural
14	NNP	proper noun, singular
15	NNPS	proper noun, plural
16	PDT	predeterminer
17	POS	possessive ending
18	PRP	personal pronoun
19	PRP$	possessive pronoun
20	RB	adverb
21	RBR	adverb, comparative
22	RBS	adverb, superlative
23	RP	particle
24	TO	to go
25	UH	Interjection
26	VB	verb, base form
27	VBD	verb, past tense
28	VBG	verb, gerund/present participle
29	VBN	verb, past participle
30	VBP	verb, sing. present, non-3d
31	VBZ	verb, 3rd person sing. present
32	WDT	wh-determiner
33	WP	wh-pronoun
34	WP$	possessive wh-pronoun
35	WRB	wh-abverb

Table B-2: List of TF-IDF Features

also	death	hate	man	past	stop	we
always	depress	have	many	people	struggle	week
an	die	he	matter	person	suck	well
and	do	head	maybe	play	suicidal	what
another	down	hell	me	please	suicide	when
any	either	help	might	point	support	where
anymore	else	her	mind	pretty	take	which
anyone	end	here	miserable	probably	talk	while
anything	enough	high	money	ready	tell	who
around	even	him	more	really	than	whole
as	ever	his	most	reason	thank	why
at	every	hope	much	right	that	will
attempt	everyone	how	music	ruin	the	wish
away	everything	hurt	my	same	their	with
back	experience	if	myself	save	them	work
bad	feel	in	need	say	then	world
be	felt	inside	never	school	there	worst
because	few	into	new	see	these	would
before	fight	it	night	seem	they	wrists
believe	find	its	no	self	thing	yeah
best	first	just	nobody	sense	things	year
better	for	keep	not	she	think	years
between	force	kid	nothing	shit	this	you
big	forever	kill	now	shoot	thoughts	your
blow	friend	kind	of	should	through	
brain	friends	know	off	show	time	
bro	from	lately	often	sleep	tire	
but	fuck	leave	oh	slit	to	
by	fun	let	ok	so	today	
call	game	life	okay	social	tonight	
can	get	like	old	some	too	
cannot	give	listen	on	someone	try	
care	glad	literally	one	something	unless	