A PRELIMINARY REPORT ON

Detecting Cyberbullying on social media using Machine Learning

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY,PUNE

IN THE PARTIAL FULFILLEMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)

SUBMITTED BY

Student Name:Harsh Agarwal	**Exam No:B150494202**
Student Name:Jagruti Jadhav	**Exam No:B150494222**
Student Name:Babita Jaybhaye	**Exam No:B150494227**
Student Name:Komal Nagar	**Exam No:B150494236**

DEPARTMENT OF COMPUTER ENGINEERING

GENBA SOPANRAO MOZE TRUST’S

PARVATIBAI GENBA MOZE COLLEGE OF ENGINEERING

SAVITRIBAI PHULE PUNE UNIVERSITY

2021-2022

CERTIFICATE

This is to certify that the project entitles

Detecting Cyberbullying on social media using Machine Learning

Submitted by

Student Name:Harsh Agarwal	**Exam No:B150494202**
Student Name:Jagruti Jadhav	**Exam No:B150494222**
Student Name:Babita Jaybhaye	**Exam No:B150494227**
Student Name:Komal Nagar	**Exam No:B150494236**

is a bonafide work carried out by Students under the supervision of **Prof. Shrikant Dhamdhere** and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of **Bachelor of Engineering** (Computer Engineering).

Prof. Shrikant Dhamdhere
Internal Guide
Dept. of Computer Engg.

Prof. Shrikant Dhamdhere
H.O.D
Dept. of Computer Engg.

External Guide

Principal
Genba Sopanrao Moze Trusts
Parvatibai Genba Moze College Of Engineering
Wagholi

Place:Pune

Date:

Abstract

With the exponential increase of social media users, cyber bullying has been emerged as a form of bullying through electronic messages. Social networks provide a rich environment for bullies to uses these networks as vulnerable to attacks against victims. Given the consequences of cyber bullying on victims, it is necessary to find suitable actions to detect and prevent it. Recently, deep neural network-based models have shown significant improvement over traditional models in detecting cyberbullying. Also, new and more complex deep learning architectures are being developed which are proving to be useful in various NLP tasks. The model is trained and evaluated on dataset that is provided by Dataturks. The dataset contained 16000 tweets gathered manually annotated by human experts. Selected Twitter-based features namely text and network-based features were used. Several classifiers are trained for determining cyberbullying

Acknowledgments

It gives us great pleasure in presenting the preliminary project report on ***‘Detecting Cyberbullying on social media using Machine Learning’****.*

I would like to take this opportunity to thank my internal guide ***Prof. Shrikant Dhamdhere*** *for giving me all the help and guidance I needed. I am really grateful to them for their kind support. Their valuable suggestions were very helpful.*

I am also grateful to ***Prof. Shrikant Dhamdhere****, Head of Computer Engineering Department, Parvatibai Genba Moze College of Engineering, Wagholi Pune-412207 for his indispensable support, suggestions.*

Harsh Agarwal
Jagruti Jadhav
Komal Nagar
Babita Jaybhaye
(B.E. Computer Engg.)

INDEX

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION OF THE PROJECT

We discuss the motivation in three parts, namely the grave nature of the menace of cyber-bullying, the algorithmic challenges in the fields of machine learning and natural language processing with respect to this problem, as well the dearth of technical solutions to tackle this problem No amount of comfort or time can fully heal the broken hearts of a parent whose child's life has either been tragically ended or has been marred because of cyber-bullying as contributing factor. Any damage done, either mentally or physically or any loss of life due to this phenomenon is frustrating mindless and is a scar upon the face of society at large One of the main motivating factors behind this work is the realization that we as computer scientists can contribute in a meaningful way towards alleviating a very serious social problem, and the dearth of work in the field of computer science in this area affords a unique opportunity to make a influencing contribution

Secondly, the computational detection of cyber-bullying raises unique questions on the many classes of algorithms in the fields of machine learning and natural language processing with respect to the phenomenon of social interaction analysis, especially in the online domain.Motivation of this project is to investigate how one might plug in the opening between these seemingly disparate fields - that an effective parameterization approach to exert the full power and weight of statistical machine learning and natural language processing involves the drawing of relevant parameters from the fields of sociology, psychiatry and sociolinguistics, all three of which have been studying the phenomenon of bullying and unkindness for decades.

Thirdly, we found it both surprising and unsettling to find a complete lack of work in the fields of computational linguistics and human-computer interaction specific to cyber-bullying.

1.2 PROBLEM DEFINITION AND SCOPE

It is important to underline the complexity of the problem of cyber-bullying and carve a crisp problem space that is ripe for the deployment of artificial intelligence and human-computer interaction paradigms. At a fundamental level, bullying amongst the young is influenced by several social and psychiatric factors. If one were to dig deeper into each such factor, it becomes abundantly clear this is a problem that is rooted in societal norms and cultures.It becomes important to define very clearly what constitutes cyber-bullying.

Cyber-bullying involves a distribution of digital harassment techniques, not limited to but involving the following: uploading of pictures or photos to embarrass a victim, stealing or hacking of personal information such as passwords and user meta information, sending or posting of abusive or damaging messages on social networking websites or through SMS text messages, sexting, making a fake account of an individual on a social network etc. In this project, we limit our work to modeling the detection of textual cyber-bullying: both explicit forms of abuse,implicit or indirect ways of abusing another person, and personal recollections of drama-related anxiety by teenagers.

1.2.1 Scope

True solutions to reduce the problem of cyber-bullying requires a fundamental restructuring of mindsets and cultural change on a huge scale. The purpose of this project is to underline the technology as an ally in mitigating its effects.Teenagers expressing recollections of distressing events can be directed to targeted help or shown messages that might reduce their difficulty.The scope of this project includes finding specific scenarios where an embedding of artificial intelligence can assist help for distressed teenagers, as upon detecting serious cases of cyber-bullying.

1.3 PROBLEM STATEMENT

Cyberbullying is a critical global issue that affects both individual victims and societies. Many attempts have been introduced in the literature to intervene in, prevent, or mitigate cyberbullying; however, because these attempts rely on the victims' interactions, they are not practical. Therefore, detection of cyberbullying without the involvement of the victims is necessary. In this problem we have to classify the statement weather if user is victim of cyberbullying or not

1.3.1 Goals and objectives

Goal and Objectives:

- Implement cyberbullying detection system using given dataset
- To study impact of various standard ml algorithms along with different data processing techniques in improving accuracy

1.3.2 Statement of scope

- Employing machine learning and interaction paradigms to provide an empathetic affordance to users is a research area that currently does not exist in the community. The future of this project is to lay broad-based principles of what that kind of paradigm it involves.

1.4 METHODOLOGIES OF PROBLEM SOLVING

1) Data collection
First, both the cyberbullying and non-cyberbullying tweets are collected from Twitter. The cyberbullying tweets are collected by retrieving Twitter with some bullying words and confirmed by crowds. The noncyberbullying tweets are collected randomly.
2) Feature extraction
After excluding unnecessary words from tweets, morphological analysis is performed. Then, textual features are extracted including n-gram, Word2Vec, Doc2Vec,emotion

values of tweets, and Twitter-specific characteristics.

3) Model generation

The collected tweets are divided into training data and test data, and the models are constructed on the training data using each type of features and each type of machine learning algorithms. The machine learning gorithms include linear models (Linear support vector machine, Logistic regression), tree-based models (Decision tree, Random forest, Gradient boosting regression tree) and deep learning models (Multilayer perceptron). In addition, cross verification and grid search are used for constructing the best model

4) Model evaluation

We evaluate how well the generated models can classify the cyberbullying and non-cyberbullying text on the test data. We use accuracy, precision, recall and F-measure as evaluation criteria

CHAPTER 2

LITERATURE SURVEY

For several years, the researchers have worked intensively on cyberbully detection to find a way to control or reduce cyberbully in Social Media platforms. Cyberbullying is troubling, as victims cannot cope with the emotional burden of violent, intimidating, degrading, and hostile messages. To reduce its harmful effects, the cyberbullying phenomenon needs to be studied in terms of detection, prevention, and mitigation.

- [1] for instance, reported how through the development of a simple language-specific method, they recorded the percentage of curse and insult words in a post, achieving a recall = 0.785 in cyberbullying identification on a small Formspring dataset.

- [2] developed a program (i.e., BullyTracer) where they identified a “cyberbullying window” 85.3 of the time (recall) and an “innocent window” 51.9 of the time in MySpace posts. More recently, the most common approach to cyberbullying detection has been through feature engineering, which has expanded the common bag-of-words representation of text by creating additional features/dimensions that use domain knowledge of linguistic cues in cyberbullying to attempt to improve a given classical classifier’s performance (e.g., Support Vector Machines - SVM, Logistic Regression). Frequent features relate to the use of profanity and how often it occurs in text

- [3] hey used seed words from three categories (abusive, violent, obscene) to calculate SO-PMI IR score and maximized the relevance of categories. Their method achieved 90 of Precision for 10 Recall. We used both of the above methods as a baselines for comparison due to similarities in used datasets and experiment settings. Unfortunately, method by [3], based on Yahoo! search engine API, faced a problem of a sudden drop in Precision

- [4] investigate the performance of several models introduced for cyberbullying detection on Wikipedia, Twitter, and Formspring as well as a new YouTube dataset. They found out that using deep learning methodologies, the performance on YouTube dataset increased

- [5] A recent paper describes similar work is that is being conducted at Massachusetts Institute of Technology. The research is aimed towards detecting cyberbullying through textual context in YouTube video comments. The first level of classification is to determine if the comment is in a range of sensitive topics such as sexuality, race/culture, intelligence, and physical attributes. The second level is determining what topic. The overall success off this experiment was 66.7accuracy for detecting instances of cyberbullying in YouTube comments. This project also used a support vector machine learner

CHAPTER 3

SOFTWARE REQUIREMENT SPECIFICATION

3.1 ASSUMPTION DEPENDENCIES

Following are assumption and dependencies mentioned for project 1. Comment should be in English language.

2. OS should support Linux application.

3. User should have web browser to use application

4.All 4 members will work for the project no option for outsource

5.Server shouldn't have any time constraint or should be greater than 10 sec

3.2 FUNCTIONAL REQUIREMENT

Following are functional requirements of system

3.2.1 Comment prediction requirement

1. The system should provide text to feature function which can take the necessary part and obtain a feature vector.

2. The system should have a well-trained SVM to generate better inputs for classifier.

3. The system should provide text parser functions which can take the whole text and separate into tokens.

4. The system needs a classifier which is well¬ trained that predicts the probability of each sentence.

3.2.2 Web page requirement

1. The system should provide a button with complete functionality. When clicked on this button, browser send the data from text box to the server.
2. The function to extract unnecessary data from web and scrap it.
3. The system should provide communication between server and client with necessary network functions.

3.2.3 Train System Requirements

The system should provide a configuration file for taking new data from admin to train models

3.3 EXTERNAL REQUIREMENTS

3.3.1 User Interface

User interface had a submit button. When user clicks submit button on a web-page it triggers the prediction function and in the text box it gives the prediction of sentence The prototype user interface is as follows 3.1.

3.3.2 Software Interfaces

In this system there will be an API named as CYB API. CYB API is used to preprocess text and for tokenization of text and predicting the outcome of sentence. This is ML API. there are two api's required one is ML-flow which generates all database and keep track of experiment metrics and dvc to create pipelines and check the track of changes in pipeline and data versioning

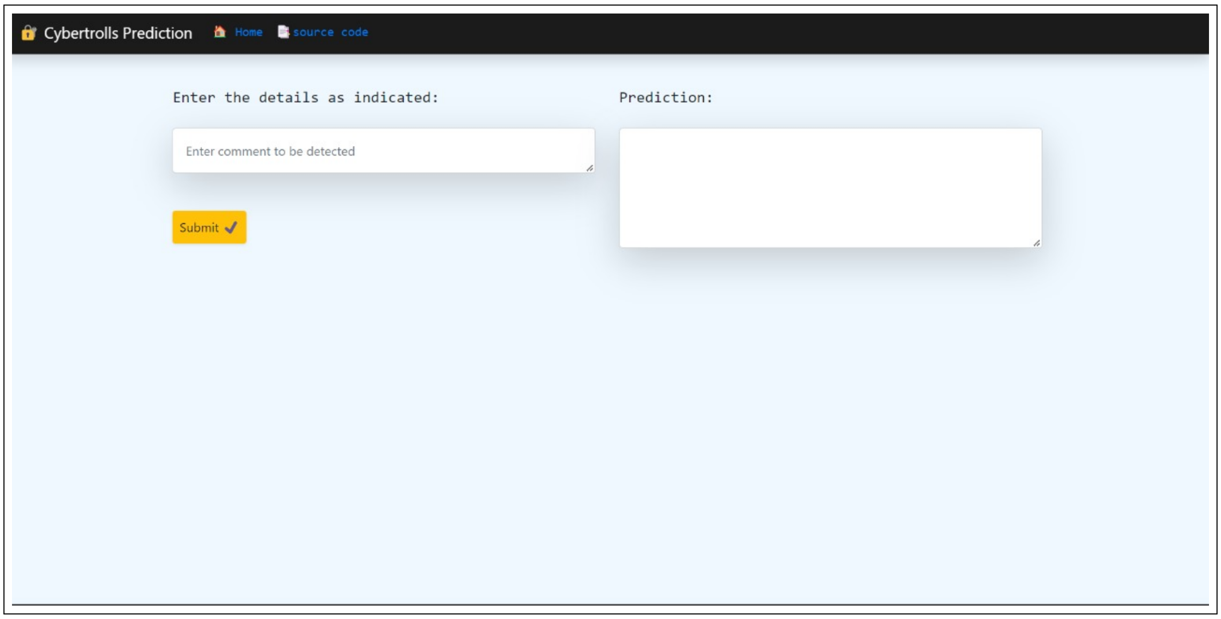


Figure 3.1: User-Interface diagram

3.3.3 Communication Interface

The communication can be done in two ways first one is between the browser and the server. Flask tool will be used to send queries and receive ones. HTTP will be used as the protocol And by using API where user had to send data in form of JSON once received by server it would return it in form of JSON only

3.4 NON FUNCTIONAL REQUIREMENTS

3.4.1 Usability

The system should be easy to use. The user should reach the prediction with one button press if possible. Because one of the software's features is timesaving.

The system also should be user friendly for admins because anyone can be admin instead of programmers.

Training the classifiers is used too many times, so it is better to make it easy.

3.4.2 Reliability

This software will be developed with machine learning, feature engineering and deep learning techniques. So, in this step there is no certain reliable percentage that is measurable.

Also, user provided data will be used to compare with result and measure reliability. With recent machine learning techniques, user gained data should be enough for reliability if enough data is obtained.

The maintenance period should not be a matter because the reliable version is always run on the server which allow users to access cyberbullying software. When admins want to update, it takes long as upload and update time of executable on server. The users can be reach and use program at any time, so maintenance should not be a big issue.

3.4.3 Performance

Calculation time and response time should be as little as possible, because one of the software's features is time saving. Whole cycle of detection of comment should not be more than 15 seconds.

The capacity of servers should be as high as possible. Calculation and response times are very low, and this comes with that there can be so many sessions at the same times. The software only used in India, then do not need to consider global sessions.

1 minute degradation of response time should be acceptable. The certain session limit also acceptable at early stages of development. It can be confirmed to user with "servers are not ready at this time" message.

3.4.4 Supportability

The system should require Python knowledge to maintenance. If any problem acquires in server side and machine learning methods, it requires code knowledge and machine learning background to solve. Client-side problems should be fixed with an update and it also require code knowledge and network knowledge.

3.5 SYSTEM REQUIREMENTS

3.5.1 Hardware Resources Required

Below table shows the hardware requirement of the software

Sr. No.	Parameter	Minimum Requirement	Justification
1	CPU Speed	2 GHz	Remark Required
2	RAM	8 GB	Remark Required

Table 3.1: Hardware Requirements
not such

3.5.2 Software Resources Required

Platform :

1. Operating System: Windows/Linux, 8 GB Ram, 2 Gb hard disk, Gpu
2. IDE: VScode
3. Programming Language: Python, Html, Css, Javascript

3.6 ANALYSIS MODEL

The developed the waterfall model development cycle for planning, Requirement Analysis, and design of the process of the development then, then we create the coding process. The implementation process will be done, the testing is done with the verification and validation process. The installation process done with the help of the Anaconda and VScode platform

3.2.

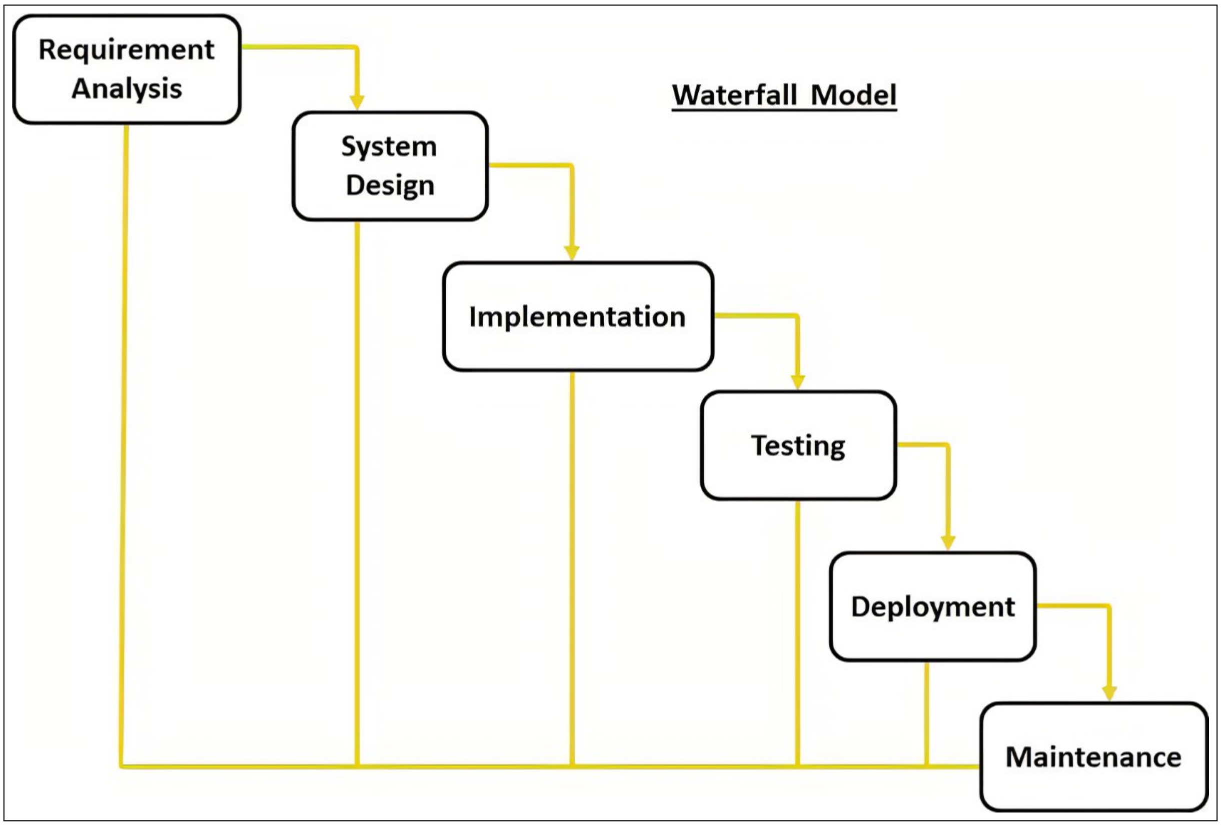


Figure 3.2: Agile model

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

A description of the program architecture is presented. each subsystem is divided into blocks. Data extraction here we just extract data from online database and convert into suitable file format in load data we convert the given file to format that can be easily processed in programming language after that we create another block to split dataset as it is important part of nlp project after that we created various mathematical model with various mathematical model in project to select best mathematical model we create log production model and this model is stored in another folder so that it can be used by prediction service to get the prediction.

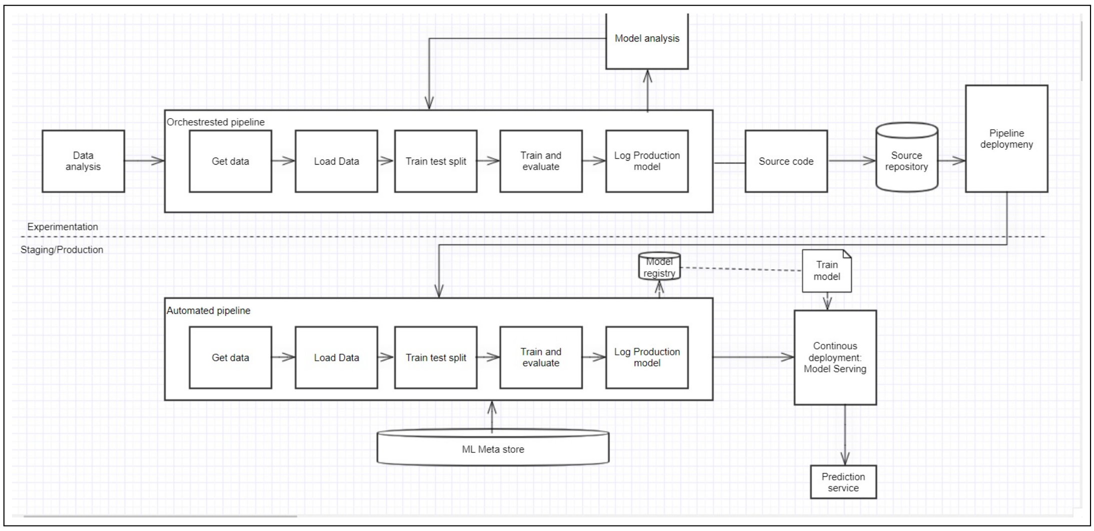


Figure 4.1: Architecture diagram

4.2 USAGE SCENARIO

4.2.1 User profiles

User: The user sends a request for the text to be checked for cyberbullying. Admin: Admin manages the website and configure a system to send responds to user requests. His/ Her another role is to maintain the algorithm and the server.

4.2.2 Use-cases

Sr No.	Use Case	Description	Actors	Assumptions
1	Webpage	Getting detection of comment	User	User click on button
2	Predict comment	Predict comment from web page	User	Error message will be displayed
3	Train System	Train the model	admin	Admin trains the classifier on new data

Table 4.1: Use Cases

4.2.3 Use Case View

Use Case Diagram. Example is given below

4.3 FUNCTIONAL MODEL AND DESCRIPTION

A description of each major software function, along with data flow (structured analysis) or class hierarchy (Analysis Class diagram with class description for object oriented system) is presented.

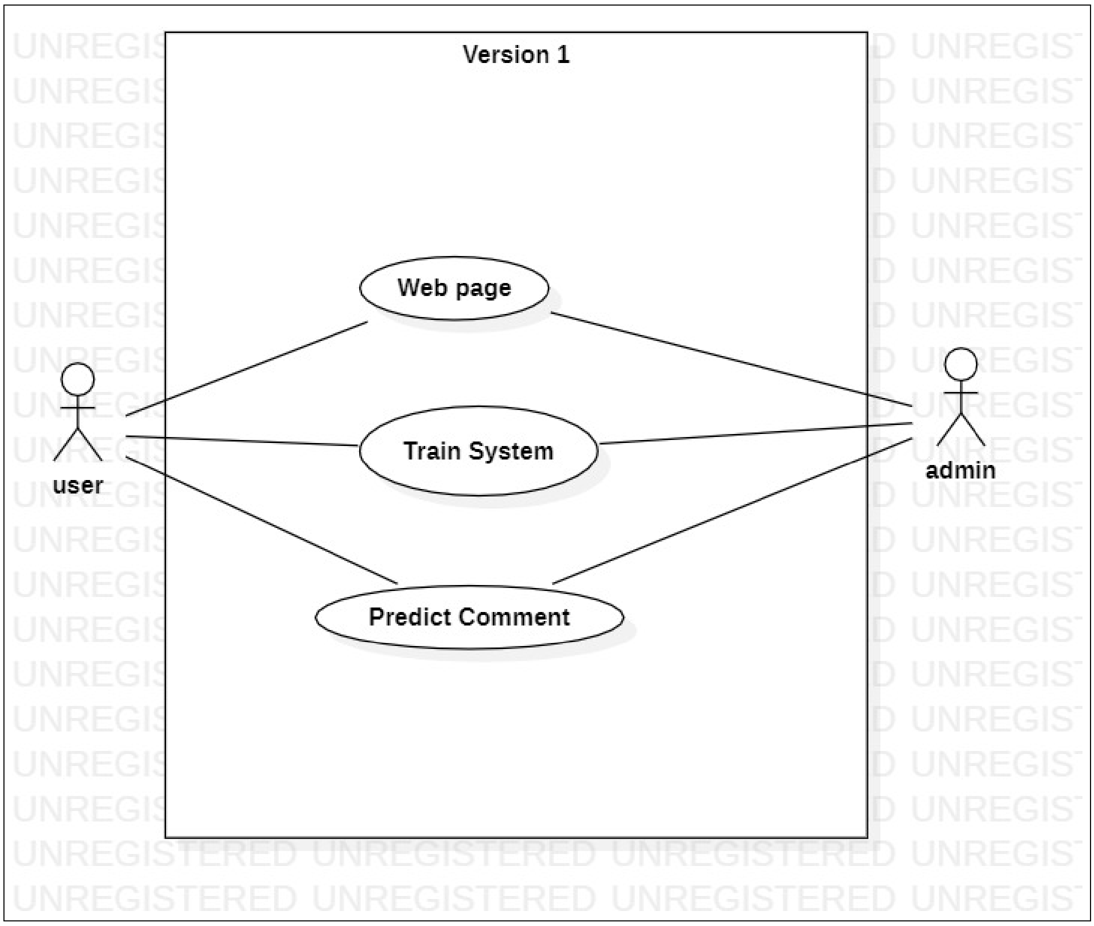


Figure 4.2: Use case diagram

4.3.1 Data Flow Diagram

4.3.1.1 Level 0 Data Flow Diagram

It's a basic overview of the whole system or process being analyzed or modeled. In DFD 0 user will fill information and system will function with help of Ml meta store.

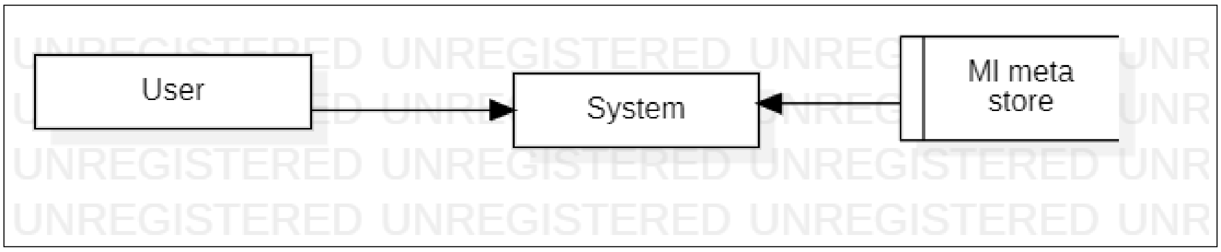


Figure 4.3: Level0: Dfd

4.3.1.2 Level 1 Data Flow Diagram

In 1-level DFD,the context diagram is decomposed into multiple bubbles/processes. Users input goes to Dataset it get processed then it will go to classification then Ml meta store would be used to give best parameter for giving accurate output.

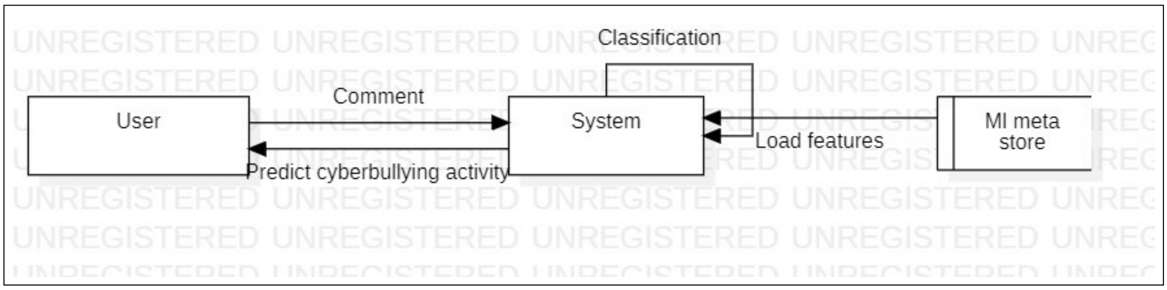


Figure 4.4: Data flow daigram-1

4.3.2 Flowchart Diagram:

The flowchart diagram shows how the algorithm start and ends using pictorial way he Flowchart describes the flow of data through an information processing systems and the parts of the flows. The flow is a set of the logic operations that meet the certain requirements. A Flowchart allows to see how the work of the process can be improved, allows to find the key elements of the process and detach the steps that are not essential or even excessive. As we can see the algorithm starts from oval shape and it takes the input from web. In second step it parse the comment then it forms word dictionary after creation of word dictionary it creates vector representation of comment after that we try to classify vector based on selected ml model
Now we try to return response from the way we recieved a response if response if form of web we return it in form of list and returning response to particular format is taken care by flask but if response recieved is in form of api then we had to return the response in form of json and besides that if there is error occured we had to create and exception and return it

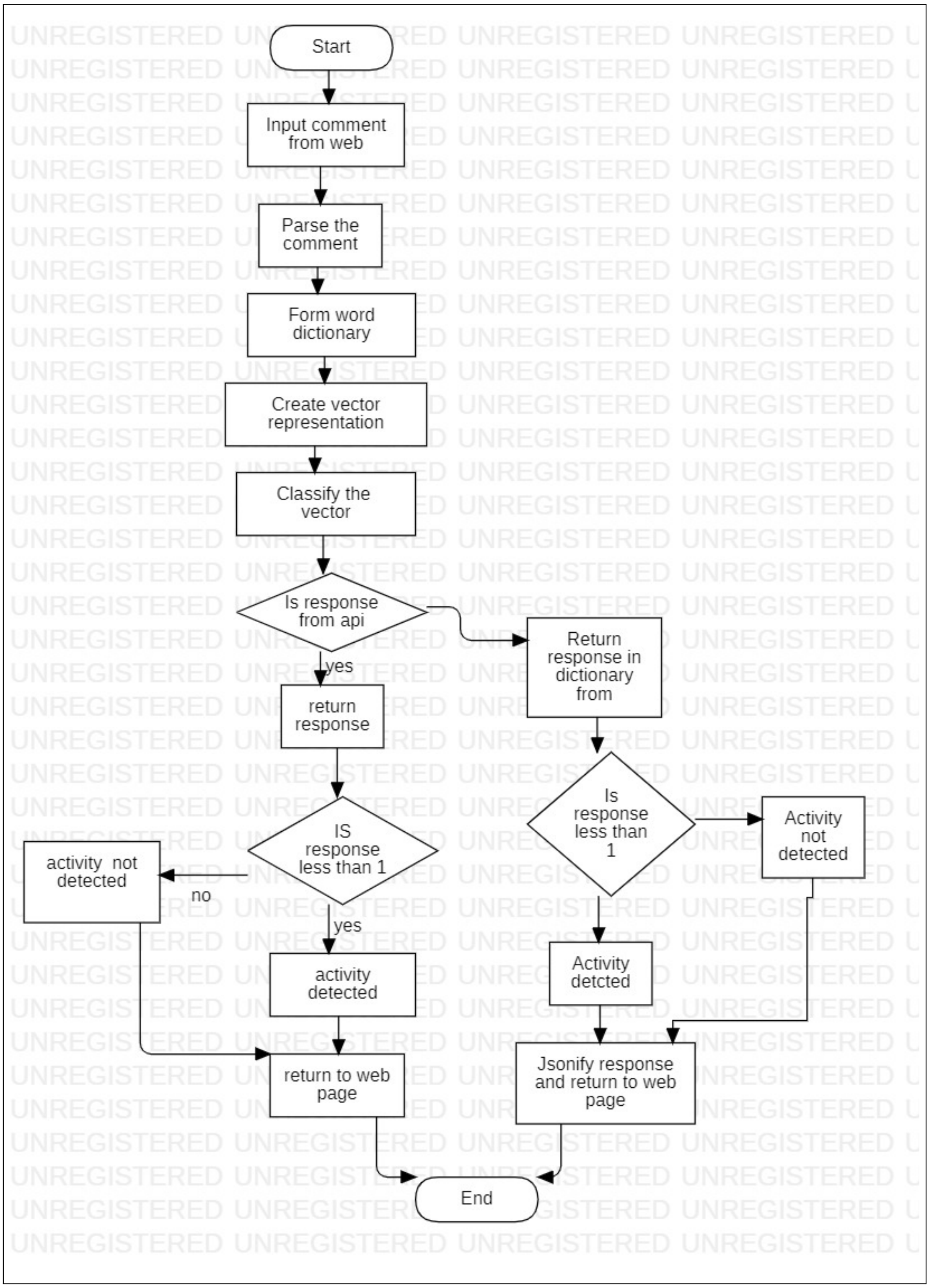


Figure 4.5: flowchart

4.3.3 Component diagram

Thus from that point of view, the below diagram shows the components that had being used in a system. These components are packages, ports, etc. It also represents the software uses text-parser, vector-creator and classifier usage at a particular moment 4.6

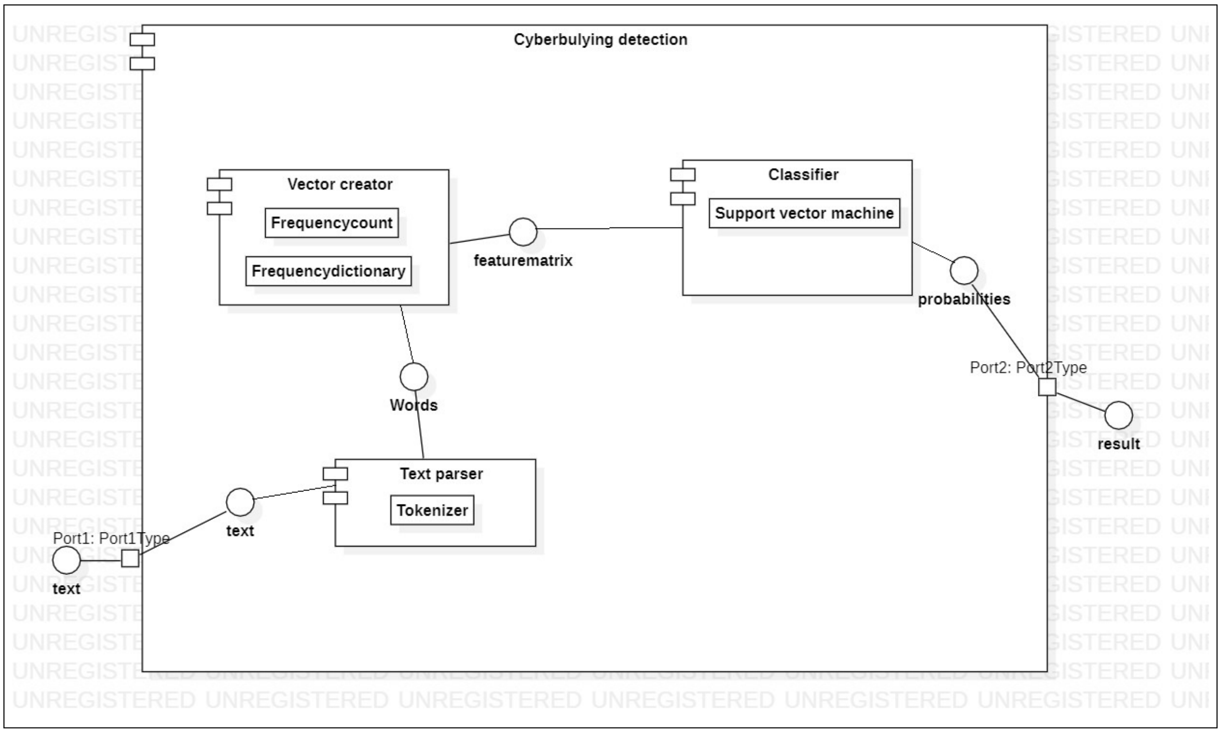


Figure 4.6: Component diagram

CHAPTER 5

PROJECT PLAN

5.1 PROJECT ESTIMATES

Use agile model and associated streams for estimation.

5.1.1 Reconciled Estimates

5.1.1.1 Cost Estimate

This cost can be calculated with the standard pay roll in India assigned to the fresher software developer according to industry standard. So let we take some amount as X per hour for development.

So cost can be calculated as below.

Number of minimum hours per developer for a month = 3 months

So Total number of hours = 3 *10*4 $\Rightarrow$ 120 Hours

So, Total cost can be said as 400XRs.

5.1.1.2 Time Estimates

Thus the total number of lines required is approximately 3.15 KLOC Efforts:

E = 3.15 * (0.903) ^ 1.02 (According to COCOMO Model)

E = 3.15 * (0.905)

E = 2.85

Development Time for Implementation and Testing

D = 2.85 Months is the development Time needed for Project.

D=Development Time for Project.

Function	Estimated KLOC
Get data	0.04
Load data	0.02
Split data	0.02
Train and evaluate	0.18
Linear Selection	0.3
Log production model	0.05
Prediction$_service$	0.15
Web app	0.243

Table 5.1: Estimation in KLOC

5.1.2 Project Resources

We required total four people which is being flexible in working with two or more roles. along with that we need open source tool i.e. python, GitHub, DVC, mlflow softwares along with that in hardware we required 8GB ram and 4 GB memory is required preferred with graphic card based on Memory Sharing, IPC, and Concurrency.

5.2 RISK MANAGEMENT W.R.T. NP HARD ANALYSIS

This section discusses Project risks and the approach to managing them.

5.2.1 Risk Analysis

The risks for the Project can be analyzed within the constraints of time and quality

5.2.2 Overview of Risk Mitigation, Monitoring, Management

Following are the details for each risk.

5.3 PROJECT SCHEDULE

This section had detailed project schedule with task set and gantt chart

ID	Risk Description	Probability	Impact		
			Schedule	Quality	Overall
1	End user resist system	Medium	High	Medium	Medium
2	Technology will not meet expectation	Medium	Low	High	Medium
3	Lack of training on tool	High	High	Medium	High
4	Staff inexperienced	Low	Low	Medium	Medium
5	Loss of Knowledge	Low	Low	Medium	Medium
6	Failure in Production	Medium	High	Low	Medium
7	Ethical and Regularity	medium	Low	Medium	Medium

Table 5.2: Risk Table

Probability	Value	Description
High	Probability of occurrence is	$> 75\%$
Medium	Probability of occurrence is	$26 - 75\%$
Low	Probability of occurrence is	$< 25\%$

Table 5.3: Risk Probability definitions [?]

Impact	Value	Description
Very high	$> 10\%$	Schedule impact or Unacceptable quality
High	$5 - 10\%$	Schedule impact or Some parts of the project have low quality
Medium	$< 5\%$	Schedule impact or Barely noticeable degradation in quality Low Impact on schedule or Quality can be incorporated

Table 5.4: Risk Impact definitions [?]

Risk ID	1
Risk Description	Technology does not meet expectation
Category	Development Environment.
Source	Software requirement Specification document.
Probability	Medium
Impact	Medium
Response	The formal meeting must be conducted
Strategy	The team must re-verify the documents and re-plan the requirement
Risk Status	identified

Risk ID	2
Risk Description	End user resist system
Category	Requirements
Source	Software Design Specification documentation review.
Probability	medium
Impact	Medium
Response	Application should be redeveloped by taking end-user in consideration
Strategy	System must be revaluated and find the reason for failure and take steps according to it
Risk Status	Identified

5.3.1 Project task set

Major Tasks in the Project stages are:

- Task 1: Is to create software environment
- Task 2: Collect data set and create pipeline
- Task 3: Model development and log production
- Task 4: Creating Web application and API development
- Task 5: Starting of test environment and create test cases

Risk ID	3
Risk Description	Lack of training on tool
Category	Technology
Source	This was identified during early development.
Probability	High
Impact	High
Response	The development team must be updated with the tools and try to regain experience
Strategy	The team manager must conduct conference to help team
Risk Status	Occured

Risk ID	4
Risk Description	development team unexperienced
Category	Technology
Source	This was identified during early development.
Probability	Low
Impact	Medium
Response	The development team must be updated with the tools and try to regain experience
Strategy	The experience team must help the weak links
Risk Status	identified

- Task 6: Create workflow and start deployment activity
- task 7: Complete heroku deployment on different branches of repositories

5.3.2 Task network

Project tasks and their dependencies are noted in this diagrammatic format begin-center

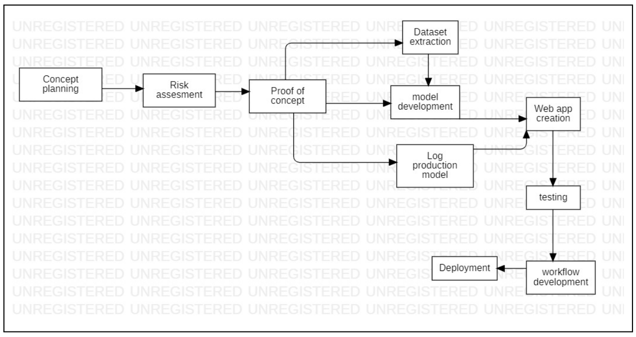
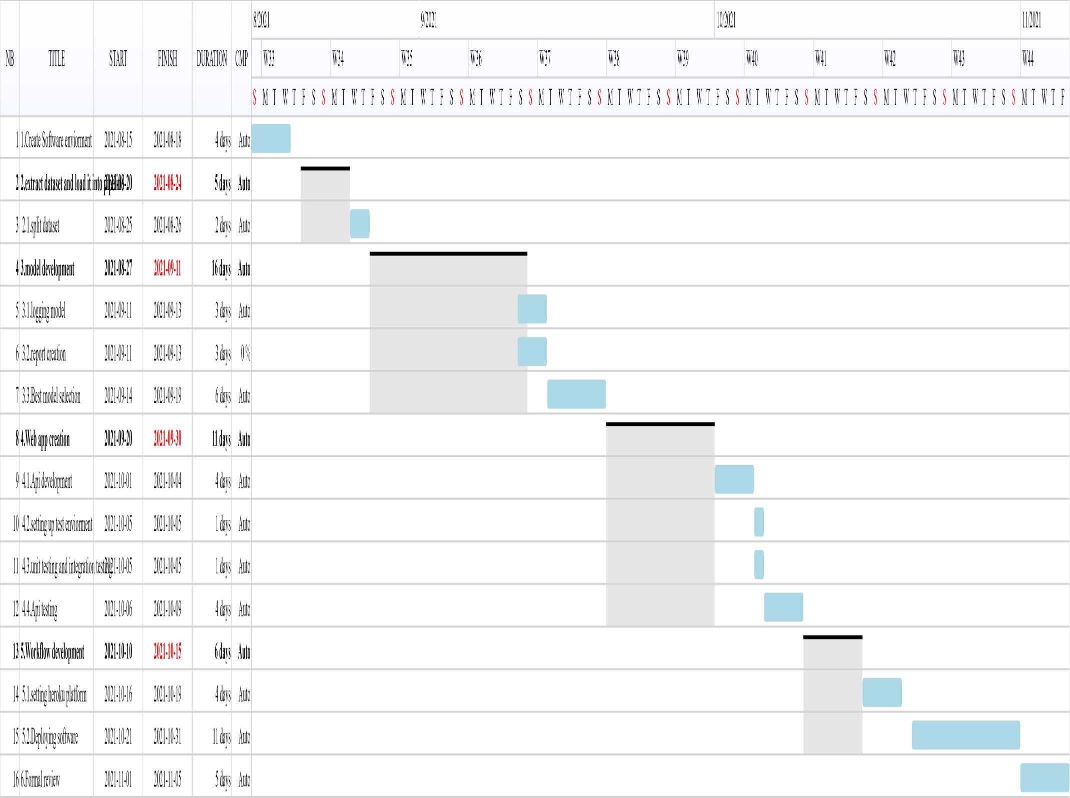


Figure 5.1: Task network

5.3.3 Timeline Chart

A project timeline chart is presented. This may include a time line for the entire project.



NB	TITLE	START	FINISH	DURATION	CMP
1	1.Create Software enviorment	2021-08-15	2021-08-18	4 days	Auto
2	**2.extract dataset and load it into pipeline**	**2021-08-20**	**2021-08-24**	**5 days**	**Auto**
3	2.1.split dataset	2021-08-25	2021-08-26	2 days	Auto
4	**3.model development**	**2021-08-27**	**2021-09-11**	**16 days**	**Auto**
5	3.1.logging model	2021-09-11	2021-09-13	3 days	Auto
6	3.2.report creation	2021-09-11	2021-09-13	3 days	0 %
7	3.3.Best model selection	2021-09-14	2021-09-19	6 days	Auto
8	**4.Web app creation**	**2021-09-20**	**2021-09-30**	**11 days**	**Auto**
9	4.1.Api development	2021-10-01	2021-10-04	4 days	Auto
10	4.2.setting up test enviorment	2021-10-05	2021-10-05	1 days	Auto
11	4.3.unit testing and integration testing	2021-10-05	2021-10-05	1 days	Auto
12	4.4.Api testing	2021-10-06	2021-10-09	4 days	Auto
13	**5.Workflow development**	**2021-10-10**	**2021-10-15**	**6 days**	**Auto**
14	5.1.setting heroku platform	2021-10-16	2021-10-19	4 days	Auto
15	5.2.Deploying software	2021-10-21	2021-10-31	11 days	Auto
16	6.Formal review	2021-11-01	2021-11-05	5 days	Auto

8/2021
9/2021
10/2021
11/2021
W33
W34
W35
W36
W37
W38
W39
W40
W41
W42
W43
W44
S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F

Figure 5.2: Timeline gantt chart

5.4 TEAM ORGANIZATION

1. Harsh Agarwal Software Developer Devops engineer
2. Jagruti Jadhav Frontend developer

3. Komal Nagar. Software developer and requirement analysis

4. Babita jaybhaye Testing and quality work

5.4.1 Team structure

The team structure comprises of various roles the roles given to members include software developer is to create backend of the web application, tester is required for some unit testing and integration testing, UI developer creates the frontend of the project communicated with software developer to create proper communication between them, Nlp engineer is required for creating nlp model and data cleaning, devops enginneer for creating pipelines and deploying it

Guide Name	Team Members
Prof. Shrikant Dhamdhare	Harsh Agarwal
	Jagruti Jadhav
	Babita Jaybhaye
	Komal Nagar

Table 5.5: Team Structure

5.4.2 Management reporting and communication

Mechanisms for progress reporting and inter/intra team communication are identified as per assessment sheet and lab time table.

CHAPTER 6

PROJECT IMPLEMENTATION

6.1 OVERVIEW OF PROJECT MODULES

Project is completely divided into modular approach first part is to create ML artifacts For creating ml artifacts we had 5 different algorithms as follows Get data set, Load data set, split data set, train and evaluate data-set and Log production model and for web app we had one module i.e prediction service

6.2 TOOLS AND TECHNOLOGIES USED

Python 3.9 installation to install the software. Point your web browser to the download page on the python website. Select the latest windows x86 Msi installer and click the link to download the msi installer, run the installer and click the next button by keeping the default setting click on next button again click yes if asked id this program should allowed to install software on your system.

SQLite Database SQLite server is relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data requested by user which may run on same computer or another computer in the network. Microsoft markets at least a dozen of different edition of SQLite version aimed at different audiences and for workload ranging from small single machine application to large internet-facing application with concurrent users

6.3 ALGORITHMS

6.3.1 Prediction Service

1. Start
2. Accept Comment
3. Process Comment
4. Create Frequency dictionary
5. Extract features
6. Used train model to get prediction
7. return prediction
8. end

6.3.2 Train test split

1. start
2. Accept Data and test ratio
3. Create a shuffle index using random library
4. Calculate test-set-size using test ratio
5. create train set and test set using slicing technique
6. End

6.3.3 Process Comment

1. start
2. Accept Comment
3. Remove stop words and perform regex operation

4. tokenize text using tweet tokenizer

5. Perform stemming

6. return comment list

7. end

6.3.4 Create frequency dictionary

1. start

2. Accept list of comments and it label

3. Initialize dictionary

4. use (word,label) as key

5. increase frequency of pair

6. end

6.3.5 Extract features

1. start

2. Accept tweet and frequency

3. Initialize vector of 1X3 dimension

4. set first term of vector as 0

5. set second term of vector represented as increment the word count for the positive label

6. set third term of vector represented as increment the word count for the negative label

7. repeat step 4 and 5 till all comments are not completed.

8. end

6.3.6 Train model

1. start
2. Accept training data
3. Assignment of target vectors(y)
4. Decide kernel function as RBF
5. Generate hyperplane
6. Maximization of margin and finding values of b and w
7. Calculation of SVM
8. Return trained model
9. end

CHAPTER 7

SOFTWARE TESTING

7.1 TYPES OF TESTING

7.1.1 Unit Testing

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input.

7.1.2 Alpha Testing

It is the most common type of testing used in the Software industry. The objective of this testing is to identify all possible issues or defects before releasing it into the market or to the user. Alpha testing is carried out at the end of the software development phase but before the Beta Testing. Still, minor design changes may be made as a result of such testing. Alpha testing is conducted at the developer's site. In-house virtual user environment can be created for this type of testing.

7.1.3 Acceptance Testing

An acceptance test is performed by the client and verifies whether the end to end the flow of the system is as per the business requirements or not and if it is as per the needs of the end user. Client accepts the software only when all the features and functionalities work as expected. It is the last phase of the testing, after which the software goes into production. This is also called User Acceptance Testing (UAT).

7.1.4 Beta Testing

Beta Testing is a formal type of software testing which is carried out by the customer. It is performed in the Real Environment before releasing the product to the market for the actual end users. Beta testing is carried out to ensure that there are no major failures in the software or product and it satisfies the business requirements from an end-user perspective. Beta testing is successful when the customer accepts the software. Usually, this testing is typically done by end-users or others. It is the final testing done before releasing an application for commercial purpose. Usually, the Beta version of the software or product released is limited to a certain number of users in a specific area. So end user actually uses the software and shares the feedback to the company. Company then takes necessary action before releasing the software to the worldwide.

7.1.5 Performance Testing

This term is often used interchangeably with ‘stress’ and ‘load’ testing. Performance Testing is done to check whether the system meets the performance requirements. Different performance and load tools are used to do this testing.

7.1.6 Security Testing

It is a type of testing performed by a special team of testers. A system can be penetrated by any hacking way. Security Testing is done to check how the software or application or website is secure from internal and external threats. This testing includes how much software is secure from the malicious program, viruses and how secure and strong the authorization and authentication processes are. It also checks how software behaves for any hackers attack and malicious programs and how software is maintained for data security after such a hacker attack.

7.1.7 White Box Testing

White Box testing is based on the knowledge about the internal logic of an application’s code. It is also known as Glass box Testing. Internal software and code working should be known for performing this type of testing. Under these tests are based on the coverage of code statements, branches, paths, conditios etc.

7.1.8 Black Box Testing

Black Box testing also known as Behavioral testing, is a software testing method in which the internal structure or design or implementation of the item being tested is not known to the tester. These test can be functional or non-functional, through usually functional. This method is named as so because the software program, in the eyes of the tester, is like a black box, inside which one cannot see. This method attempts to find error like incorrect or missing functions, interface error, behavior or performance error etc.

7.1.9 Regression Testing

Testing an application as a whole for the modification in any module or functionality is termed as Regression Testing. It is difficult to cover all the system in Regression Testing, so typically automation testing tools are used for these types of testing.

7.1.10 System Testing

Under System Testing technique, the entire system is tested as per the requirements. It is a Black-box type testing that is based on overall requirement specifications and covers all the combined parts of a system.

7.1.11 Smoke Testing

Whenever a new build is provided by the development team then the software testing team validates the build and ensures that no major issue exists. The testing team ensures that the build is stable and a detailed level of testing is carried out further. Smoke Testing checks that no show stopper defect exists in the build which will prevent the testing team to test the application in detail. If testers find that the major critical functionality is broken down at the initial stage itself then testing team can reject the build and inform accordingly to the development team. Smoke Testing is carried out to a detailed level of any functional or regression testing.

7.1.12 Integration Testing

Integration testing is testing in which a group of components are combined to produce output. Also, the interaction between software and hardware is tested in integration testing if software and hardware components have any relation. It may fall under both white box testing and black box testing. It has two testing under it. These are- (a) Top to bottom. (b) Bottom to top. (a) Top to bottom:In this, the system is divided into different modules. Each and every module is tested from top to bottom. (b) Bottom to top: In this type of testing, every module is tested individually and at the end all modules are integrated.

7.2 TEST CASES

Sr. No.	Test Case	User Input	Expected Result	Actual Result	Status
1	Form Response	Comment	value between 0 and 1	0	Pass
2	Api Response	Comment	value between 0 and 1	1	Pass

Table 7.1: Test cases

For testing test cases we had created s separate test environment using tox which requires setup file which creates environment for testing this helps to automate test. We had used pytest for testing system.

CHAPTER 8

RESULTS

8.0.1 Outcome

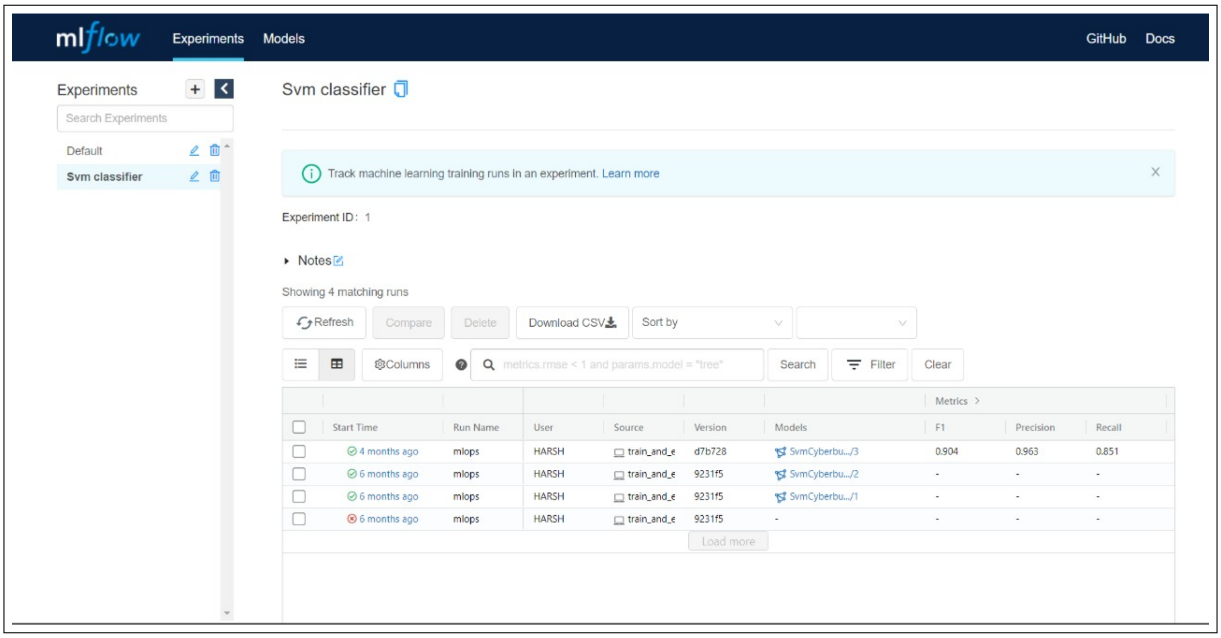
The user would be able to identify if comment contains any cyber bullying activity or not if it is detected the system can block that user eventually.

Paper	Accuracy	Precision	F1-Score	Recall
Base paper 1	71.25	71	71	70
Base paper 2	81.7	-	-	-
Base paper 3	-	92	93	-
Experiment Result	92.9	96.3	90.04	85.1

Table 8.1: Result comparison with respect to base paper

8.1 SCREENSHOTS

Following diagram shows the training of model and how it had being logged in software 8.1.



Svm classifier

Track machine learning training runs in an experiment. Learn more

Experiment ID : 1

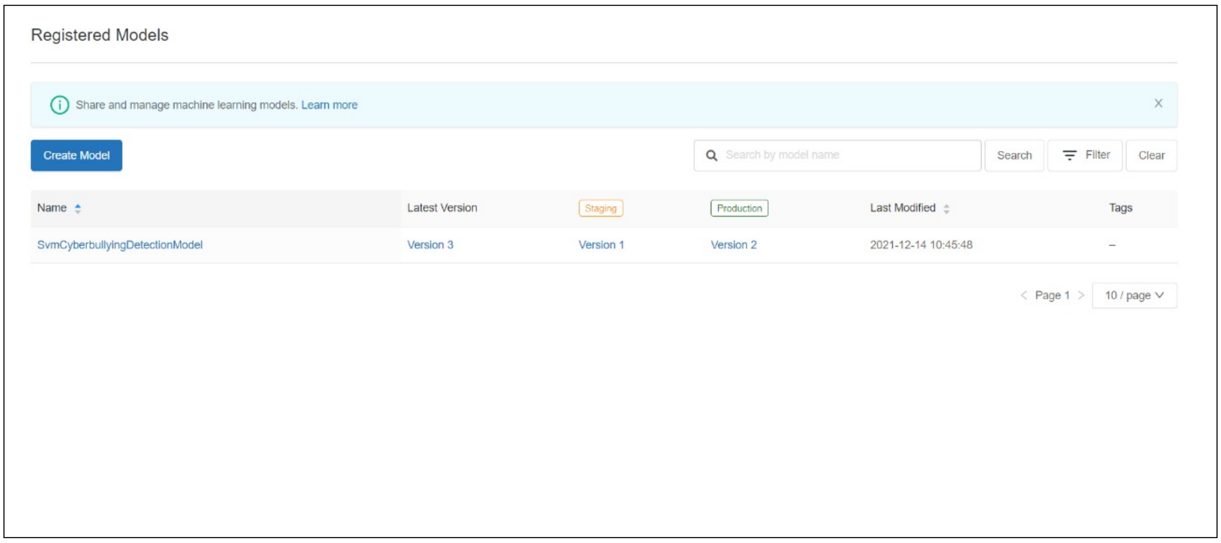
Notes

Showing 4 matching runs

metrics.rmse < 1 and params.model = "tree"

							Metrics >		
□	Start Time	Run Name	User	Source	Version	Models	F1	Precision	Recall
□	✓ 4 months ago	mlops	HARSH	train_and_e	d7b728	SvmCyberbu.../3	0.904	0.963	0.851
□	✓ 6 months ago	mlops	HARSH	train_and_e	9231f5	SvmCyberbu.../2	-	-	-
□	✓ 6 months ago	mlops	HARSH	train_and_e	9231f5	SvmCyberbu.../1	-	-	-
□	☒ 6 months ago	mlops	HARSH	train_and_e	9231f5	-	-	-	-

Figure 8.1: Diagram showing result of model



Registered Models

Share and manage machine learning models. Learn more

Name	Latest Version	Staging	Production	Last Modified	Tags
SvmCyberbullyingDetectionModel	Version 3	Version 1	Version 2	2021-12-14 10:45:48	–

Figure 8.2: Following figure show the status of the model which model is serving and which model is currently in staging area where staging and production model depends on accuracy of model

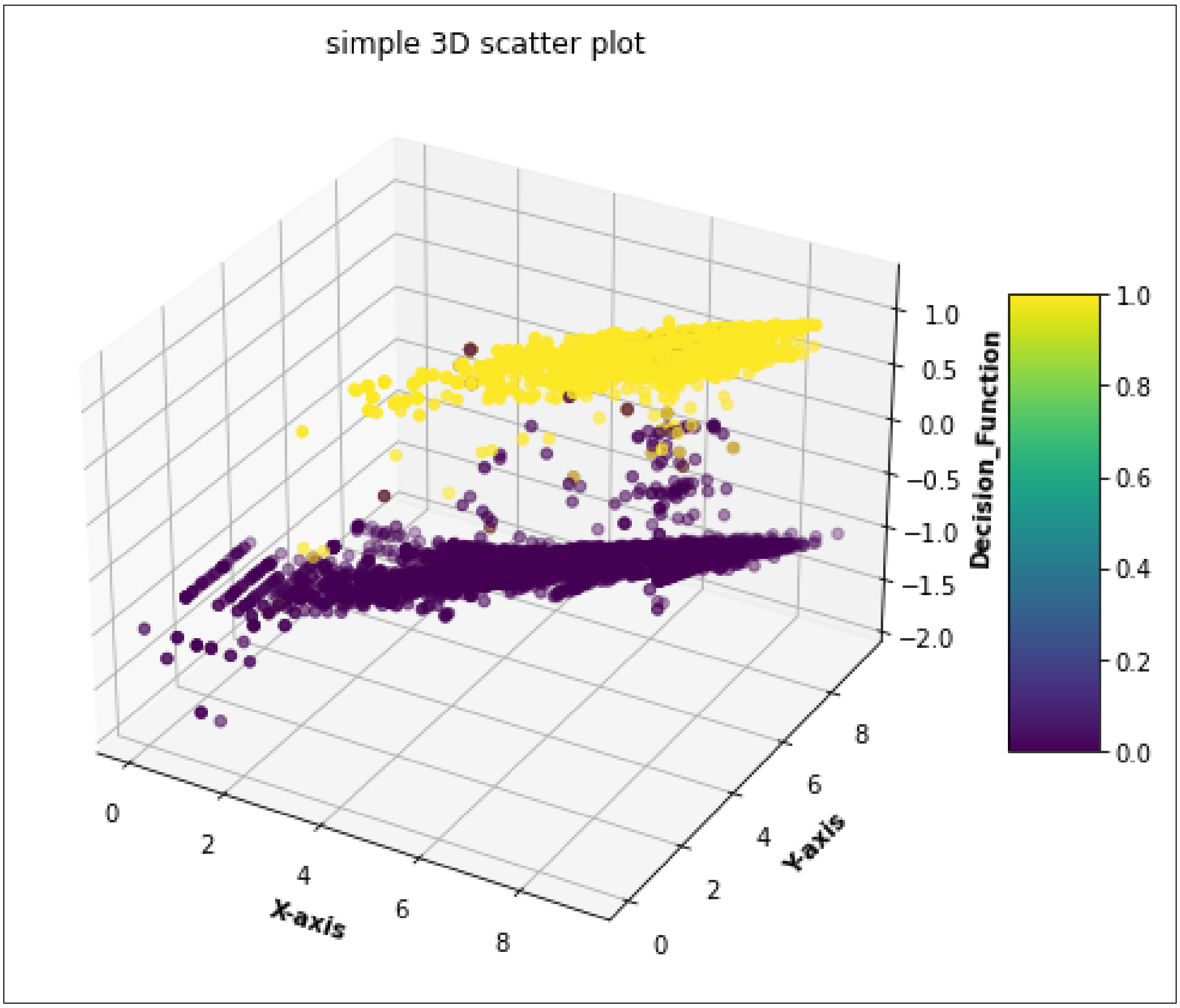
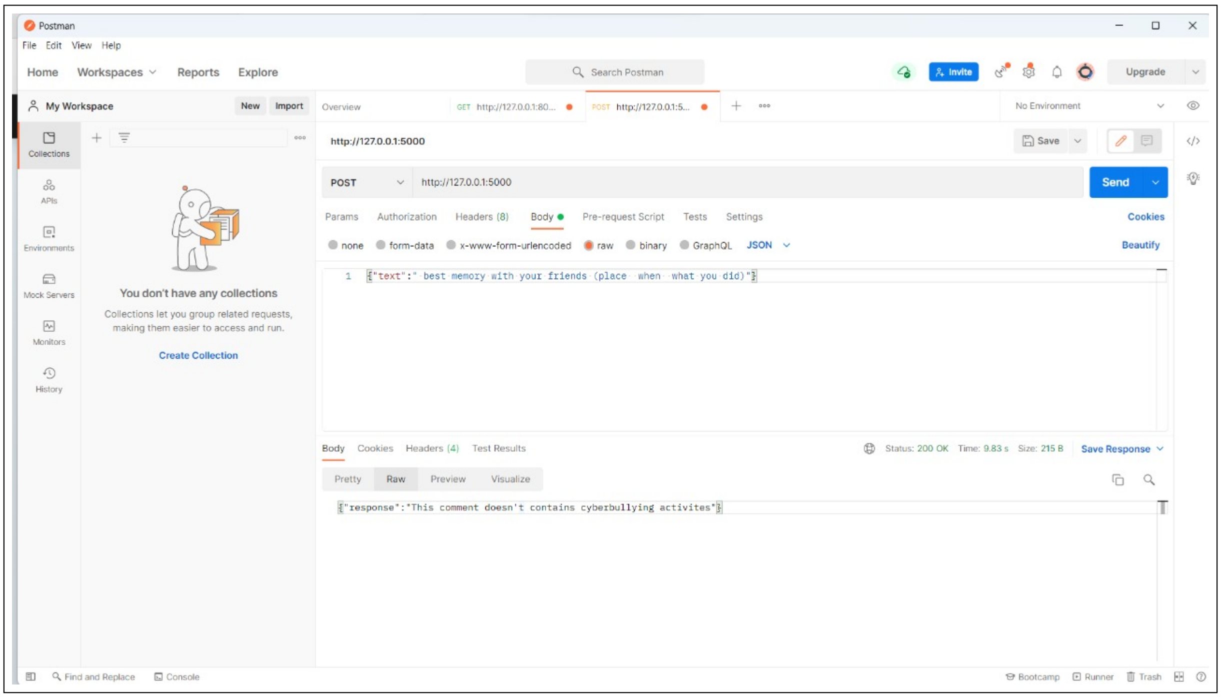


Figure 8.3: Figure represents how the machine learning model is able to classify the training instance where yellow



http://127.0.0.1:5000

POST http://127.0.0.1:5000

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{"text":" best memory with your friends (place  when  what you did)"}
```

Body — Status: 200 OK Time: 9.83 s Size: 215 B

```
{"response":"This comment doesn't contains cyberbullying activites"}
```

Figure 8.4: Figure represents the testing and result of api using postman app

CHAPTER 9

CONCLUSION

9.1 CONCLUSIONS

We have proposed the method to automatically detect cyberbullying text . Multiple textual features and multiple machine learning algorithms are used to construct classification models.With the experiments based on the collected comments, the quality of automatic cyberbullying detection is evaluated and the best model performs over 90% for the four criteria: Accuracy precision recall F-measure. Extracting new bullying words from the current comments and using them to obtain new comments iteratively. A more effective way to handle slang text in non-English languages can also be considered to improve the representations of such words. With the world coming more closer in this new digital age, machine learning models must be also able to handle diverse data effectively. Maintaining privacy and liberty of all users across cyberspace is essential

9.2 FUTURE SCOPE

Future Scope of this project would be working on detection of sarcastic comments and to identify cyberbullying activities in image and video form

9.3 APPLICATIONS

User can freely express their thoughts on social media.Users will be free from social-anxiety.Users would less frequently delete post on social media and loss fear of trolling. Along with that this can be easily integrated with various application with its feature of API there is no need to explicitly generate key due to which it can be useful for both mobile as well as web app.

CHAPTER 10

REFERENCES

[1] K. Reynolds, A. Kontostathis, and L. Edwards, "Using machine learning to detect cyberbullying," in *2011 10th International Conference on Machine learning and applications and workshops*, vol. 2, pp. 241–244, IEEE, 2011.

[2] J. Bayzick, A. Kontostathis, and L. Edwards, "Detecting the presence of cyberbullying using computer software," 2011.

[3] T. Nitta, F. Masui, M. Ptaszynski, Y. Kimura, R. Rzepka, and K. Araki, "Detecting cyberbullying entries on informal school websites based on category relevance maximization," in *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pp. 579–586, 2013.

[4] R. R. Dalvi, S. B. Chavan, and A. Halbe, "Detecting a twitter cyberbullying using machine learning," in *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 297–301, IEEE, 2020.

[5] K. Dinakar, R. Reichart, and H. Lieberman, "Modeling the detection of textual cyberbullying," in *fifth international AAAI conference on weblogs and social media*, 2011.

ANNEXURE A

ALGORITHMIC DESIGN

A.1 FEASIBILITY ASSESSMENT

Feasibility study is the test of a system proposal according to its workability, impact on the organization, ability to meet user needs, and effective use of resources. It focuses on the evaluation of existing system and procedures analysis of alternative candidate system cost estimates. Feasibility analysis was done to determine whether the system would be feasible. The development of a computer based system or a product is more likely plagued by resources and delivery dates. Feasibility study helps the analyst to decide whether or not to proceed, amend, postpone or cancel the project, particularly important when the project is large, complex and costly.

Once the analysis of the user requirement is complement, the system has to check for the compatibility and feasibility of the software package that is aimed at. An important outcome of the preliminary investigation is the determination that the system requested is feasible.

A.1.1 Technical Feasibility

This project is developed by using the Java Programming language which is a best application oriented programming language. The Proposed model uses open source IDE like Netbeans and not including any licenses. So the Project doesn't need any special permissions or certificates to deploy and technically it doesn't depend on any further software's. So its feasibility is tested and deployed in many controller scenarios successfully.

A.1.2 Economical Feasibility

The proposed model is developed using open source IDE and dataset Server like Netbeans and Syntactical Dataset. And again the proposed model is not using any licensed tools or API to develop the model so while developing the project there is no need of investments of funds.

A.1.3 Deployment Feasibility

Installation manual of software can provide to the end user with all easily available open source development kit. And it takes less than an hour of time to do all the setup for even an untrained person.

A.1.4 Time Feasibility

A time feasibility study will take into account the period in which the project is going to take up to its completion. A project will fail if it takes too long to be completed before it is useful.

A.2 NP HARD

A.2.1 Computations, Decisions and Languages

The most common resource to analyze software is time and number of execution steps, this is generally computed in terms of n. We will use an informal model of a computer and an algorithm. All the definitions can be made precise by using a model of a computer such as a Turing machine.

While we are interested in the difficulty of a computation, we will focus our hardness results on the difficulty of yes -no questions. These results immediately generalize to questions about general computations. It is also possible to state definitions in terms of languages,where a language is defined as a set of strings: the language associated with a question is the set of all strings representing questions for which the answer is Yes.

A.2.2 The Class P

The collection of all problems (Algorithms or methods that we are using in our project) that can be solved in polynomial time is called P. That is, a decision question is in P if there exists an exponent k and an algorithm for the question that runs in time O(n^k) where n is the length of the input.

P roughly captures the class of practically solvable problems. Or at least that is the conventional wisdom. Something that runs in time 2^n requires double the time if one adds one character to the input. Something that runs in polynomial time does not suffer from this problem.

A.2.3 The Class NP

The collection of all problems that can be solved in polynomial time using non determinism is called NP. That is, a decision question is in NP if there exists an exponent k and an non deterministic algorithm for the question that for all hints runs in time O(n^k) where n is the length of the input.

A.2.4 P versus NP

It would seem that P and NP might be different sets. In fact, probably the most important Unsolved problems in Mathematics and Computer Science today is:

Conjecture. P $\neq$ NP

If the conjecture is true, then many problems for which we would like efficient algorithms do not have them. This would be sad. If the conjecture is false, then much of cryptographic under threat. Which would be sad.

A.2.5 NP Complete

While we cannot determine whether P = NP or not, we can, however, identify problems that are the hardest in NP. These are called the NP-complete problems. They have the property that if there is a polynomial-time algorithm for any one of them then there is a polynomial-time algorithm for every problem in NP.

A.2.5.1 Definition

A decision problem S is defined to be NP-complete if

a) It is in NP; and

b) For all A in NP it holds that A≤PS.

A.2.5.2 Note that this means that

- If S in NP-complete and S in P, then P=NP.
- If S is NP-complete and T in NP and S≤PT, then T is NP-complete.

A.2.5.3 Example

We can state this even in simpler form, like as shown below:

Let us consider a module A_{NN} (Artificial Neural Network) in our system called S,

Then If A_{NN} is set to change in time T

$A_{NN}{}'$(changed module) will be the changed module

If $(A_{NN}{}' \in S) \leq T$

Then system is considered as NP Complete. Our System unconditionally satisfies this problem, So we can conclude our system as NP Complete.

A.3 MATHEMATICAL DESIGN:

Let S be closed system defined as S=Ip,Op,Ss,Su,Fi,A
To select the input from the system and perform various action from the set of action A and the state Su be attained.
S=Ip,Op,Ss,Su,Fi,A
Where, IP1=Comment
IP2=Data-File
Set of actions=F1,F2,F3,F4
Where
F1=Preprocessing

F2=Classification

F3=Analysis

F4=Cyberbullying activity Detection

S=set of users

Ss=rest state, form state, Api state, Data state

Su=Success state is successful response

1)Input1:IP1=Comment

2)Input2:IP2:Commentjson

3)Input3:IP3:Data File

1)Output1:Op1:Classification and analysis

2)output2:Op2:Train model

ANNEXURE B

REVIEWERS COMMENTS OF PAPER SUBMITTED

B.1 PAPER SUBMITTED

Cyberbullying Detection On Social Media Using Machine Learning

Harsh Agarwal Jagruti Jadhav Komal Nagar Babita Jaybhaye Shrikant Dhamdhere

Parvatibai Genba moze college of engineering,Pune-411001,India
Parvatibai Genba moze college of engineering,Pune-411001,India
Parvatibai Genba moze college of engineering,Pune-411001,India
Parvatibai Genba moze college of engineering,Pune-411001,India
Parvatibai Genba moze college of engineering,Pune-411001,India

ABSTRACT

The popularity of online social networks has created massive social interaction among their users, and this leads to a huge amount of user-generated communication data. In recent years, Cyberbullying has grown into a major problem with the growth of online interaction and social media. Cyberbullying has been recognized recently as a serious national health concern among online social network users and developing an efficient detection model holds tremendous practical significance. In this paper, we have proposed set of unique features derived from Twitter based on these features, we developed a supervised machine learning solution for detecting cyberbullying in the Twitter. An evaluation demonstrates that our developed detection model based on our proposed features, achieved results with an area under the precision of 0.963 and an f-measure of 0.904. These results indicate that the proposed model based on these features provides a feasible solution to detecting Cyberbullying in online interacting environments

Keywords: cyberbullying; machine learning; classifiers; Naive Bayes; support vector machine (SVM); cyber-aggressive;supervised; cybercrime; online-communication;twitter

1. Introduction

Cyberbullying is a deliberate and repetitive act to harm or humiliate someone using information and communication technologies such as mobile phones, emails and social media [1] [2]. It is often categorized into various forms, such as cyber harassment (i.e. repetitively harassing and threatening someone), denigration/slandering (i.e. sharing false information about someone), flaming (i.e. brief insulting online interactions) and happy slapping (i.e. recording a session while a person is being bullied for circulation purpose), among others [2]. Impacts of cyberbullying are detrimental in nature, ranging from emotional (anger, fear, self-blame etc.) to psychological (low self-esteem, depression, suicidal etc.) and physical (loss of sleep, headache, eating disorder etc.).

Despite the various prevention and intervention strategies, cyberbullying perpetration has not decreased in the last one decade [3]. Recent studies have looked into automatically detecting cyberbullying incidents, for instance, an affect analysis based on a lexicon and Support Vector Machine (SVM) was found to be effective in detecting cyberbullying, however the accuracy decreased when the size of the data increased, suggesting that SVM may not be ideal in dealing with frequent language ambiguities typical for cyberbullying [4]. [5] automatically collected data from an in-game chat (World of Tanks) and found cyberbullying to be a learned behaviour (i.e. new players are less likely to engage in cyberbullying).

Cyberbullying and its impact on social media: Cyberbullying is not just limited to creating a fake identity and publishing/posting some embarrassing photo or video, unpleasant rumours about someone but also giving them threats. The impacts of cyberbullying on social media are horrifying, sometimes leading to the death of some unfortunate victims. The behaviour of the victims also changes due to this, which affects their Emotions, self-confidence and a sense of fear is also seen in such people.

2. Background

Cyberbullying – a social media problem:

The use of information and communication technologies, particularly social media has revolutionized the manner in which people communicate and form relationships with one another, with statistics around the world indicating a high prevalence rate of social media applications. For instance, according to the recent report by Pew Research Centre (2018), Instagram (75%) and Snapchat (73%) were found to be most popular among those between 18 and 24 years old whereas Facebook and YouTube were more popular among those older than 50 (i.e. 68%). This unfortunately, provides an avenue for anti-social behaviours such as misogyny [6] [7], sexual predation sexism [8] and cyberbullying perpetration [9] [10] [11]. Facebook, for instance, is one of the most popular social

Facebook, for instance, is one of the most popular social media platforms that allows its users to create their own profiles, upload their photos and videos, and send messages (both private and public). It has a wide reach, as any comments or posts can reach thousands of people, especially through "liking" and "sharing" mechanisms, and thus allowing cyberbullies to distribute nasty or unwanted information about their victims easily [12]. Instagram allows its users to share photos and videos, to follow others and support Stories. Like 7 Facebook, it is also easy for one to set up new, anonymous profiles for cyberbullying perpetration. The velocity and size of the distribution mechanism allow hostile comments or humiliating images to go viral within hours [12].

1. ***Features***

This section specifically focuses on the features incorporated in our cyberbullying detection model. It encompasses user personalities focusing on Big Five and

Dark Triad models, sentiment, emotion and Twitter-based features.

2. ***User personalities***

One of the most comprehensive and popular method to determine personality is based on the Big Five model [13] [14], which is a hierarchical organization of personality traits in terms of five basic dimensions/facets: Extraversion - the tendency of being outgoing, sociable, to be interested in other people, assertive, active, paying more attention to external events and excitement seeking Agreeableness - the tendency to be kind, friendly, gentle, getting along with others and being warm to other people Conscientiousness – it presents how much a person pays attention to others when making decisions Neuroticism - the tendency to be depressed, fearful and moody Openness - the tendency to be creative, perceptive, thoughtful, broad-minded, and willing to make adjustments in activities in accordance with new ideas.

3. ***Cyberbullying detection and machine learning***

Machine learning, an application of artificial intelligence provides systems the ability to automatically learn and improve from experience without being explicitly programmed, often differentiated as supervised, unsupervised or semi-supervised algorithms [15].The supervised algorithms take a set of training instances to build a model that generates a desired prediction for an unseen instance (i.e. based on labelled/annotated data), whereas unsupervised algorithms do not depend on labelled data, and thus often used for clustering problems [15] As cyberbullying is deemed to be a classification problem (i.e. categorizing an instance as bully or non-bully), the supervised learning algorithms were adopted in the present study.

Studies on cyberbullying detection are mainly based on superintending algorithms such as Naïve Bayes, SVM, Decision Trees (J48), and Random Forest, often with performance comparisons made between several of these classifiers [10] [16]. Naïve Bayes is a Bayesian theorem algorithm and is well-known for its ability to classify texts based on a probability (i.e., outcomes are based on the highest probability). It is therefore, wellsuited for real-time predictions, text classifications and recommendation systems [17]. It assumes independence between predictors, that is, the presence/absence of a feature is unrelated to the presence/absence of any other feature. Therefore, in the context of tweets, each word or feature is considered as a unique variable by Naive Bayes to determine the probability of that word/feature. For instance, [18] proposed a model for detecting cyberbullying using Na¨ıve Bayes, whereby the presence of an offensive word indicates cyberbullying, and the absence indicates otherwise. The authors however, did not evaluate their proposed model, but other similar studies such as [19] reported an overall accuracy of 63% using Naïve Bayes to detect cyberbullying based on YouTube comments.

The present study adopted a similar approach whereby the presence of specific features (or combination of features) (e.g., high number of followers-following or a negative personality) may result in a tweet to be classified as a bully. J48 is a popular decision tree algorithm, which uses the depth-first strategy that considers all the possible tests to split the dataset before one with the highest information gain is selected [20]. The trees contain several nodes, that is, root (main node, no incoming edges), internal (with incoming and outgoing edges) and leaf (no outgoing edges). Both the root and internal nodes correspond to each feature tested whereas the leaf node is the final classification. Therefore, in the context of cyberbullying, features such as number of followers, popularity, positive sentiment can be used as the root or internal nodes, whereas bully or not-bully will be the corresponding leaf node. J48 is generally easy to use and relatively fast, however the preparation of large decision trees (i.e., large datasets with many features) are complex and time-consuming (Zhao and Zhang, 2008). [16] explored the social network graphs features, namely the relationships between users and related features (e.g., number of friends), and network embeddedness (i.e., relationship between users) etc. using J48, with results indicating an accuracy of 62.8

3. Related work

A previous study proposed an approach for offensive language detection that was equipped with a lexical syntactic feature and demonstrated a higher precision than the traditional learningbased approach [21]. A YouTube databased study [22] applied SVM to detect cyberbullying, and determined that incorporating userbased content improved the detection accuracy of SVM. Using data sets from Myspace, developed a genderbased cyberbullying detection approach that used the gender feature in enhancing the discrimination capacity of a classifier [23]. included age and gender as features in their approach; however, these features were limited to the information provided by users in their online profiles [24]. Moreover, most studies determined that only a few users provided complete information about themselves in their online profiles. Alternatively, the tweet contents of these users were analysed to determine their age and gender [23]; [25]. Several studies on cyberbullying detection utilized profane words as a feature, thereby significantly improving the model performance. A recent study [26] proposed a model for detecting cyberbullies in Myspace and recognizing the pairwise interactions between users through which the influence of bullies could spread. Nalini and Sheela proposed an approach for detecting cyberbullying messages in Twitter by applying a feature selection weighting scheme [27]. included pronouns, skip-gram, TFeIDF, and N-grams as additional features in improving the overall classification accuracy of their model [28].

4. Materials and methods

1. Experimental setup

Stepwise Procedure of SVM utilized in detecting the cyberbullying Steps:

1. For a particular location, a limited number of tweets will be fetched through dataset

2. The Data Pre-processing, Data Extraction will be performed on the fetched Tweets

3. Pre-processed tweets will be passed to SVM and Naïve Bayes model (see Developing the Model section) to calculate the probabilities of fetched tweets to check whether a fetched tweet is bullying or not.

4. If the probability of fetched tweet lies in the range of 0.5 to 1, then the tweet will not be considered as a bullied tweet.

5. Again, the list of tweets will be passed to the SVM and Naive Bayes model to predict the results of the tweets.

6. And again, the average probability of that tweet will be calculated and if it lies above 0.5 then it will be considered as a bullied tweet and it will be recorded

in our database. If the average probability is less than 0.5 then the record will be removed from the database.

2. ***Developing the model***

The entire model is divided into 3 major steps: Pre-processing, the algorithm, and feature extraction.

Pre-processing

The Natural Language Toolkit (NLTK) is used for the pre-processing of data. NLTK is used for tokenization of text patterns, to remove stop words from the text, etc. Tokenization: In tokenization, the input text is split as the separated words and words are appended to the list. Firstly, TweetTokenizer is used to tokenized text into the sentences. Then 4 different tokenizers are used to tokenize the sentences into the words:

o TweetTokenizer

o WordPunctTokenizer

o TreebankWordTokenizer

o PunctWordTokenizer

Lowering Text: It lowers all the letters of the words from the tokenization list. Example: Before lowering “Hey There” after lowering “hey there”. Removing Stop words: This is the most important part of the pre-processing. Stop words are useless words in the data. Stop words can be get rid of very easily using NLTK. In this stage stop words like -https, -, are removed from the text.

Wordnet lemmatize: Wordnet lemmatize finds the synonyms of a word, meaning and many more and links them to the one word

Feature Extraction:

In this step, the proposed model has transformed the data in a suitable form which is passed to the machine learning algorithms. The frequency dictionary is used to extract the features of the given data. Features of the data are extracted and put them in a list of features. Also, the frequency of word defining polarity (i.e., the text is Bullying or Non-Bullying) of each text is extracted and stored in the list of features

Algorithm Selection:

To detect social media bullying automatically, supervised Binary classification machine learning algorithms like SVM with linear kernel and Naive Bayes is used. The reason behind this is both SVM and Naive Bayes calculate the probabilities for each class (i.e. probabilities of Bullying and Non-Bullying tweets). Both SVM and NB algorithms are used for the classification of the two-cluster. Both the machine learning models were evaluated on the same dataset. But SVM outperformed Naive Bayes of similar work on the same dataset. Classification report [9] is also evaluated. The accuracy, recall, f-score, and precision are also calculated

Precision = TP / (TP+FP)

Recall =TP/(TP+FN)

F-Score = 2*(Precision*Recall) / (Precision + Recall)

Where ,

TP = True positive numbers

TN = True negative numbers

FN = False negative numbers

FP = False positive numbers

Support Vector Machine

Support Vector Machine is a supervised classification machine learning algorithm. SVM can be used for both regression and classification. SVM also calculates the probabilities for each category. SVM with non-Linear Kernel is used as our data is linearly separable.

HYPERPLANE: The main aim of the SVM is to find the hyperplane which divides the dataset into two categories. Many hyperplanes separate two categories of the data points. The main aim of the SVM is to find the hyperplane with a maximum margin. For 2 attributes hyperplane is just a line. As the number of features increases, it is very difficult to imagine the hyperplanes' dimension. In our model, as there are only 2 classes, i.e., Bullying and Non-Bullying hyperplane was just a line. SUPPORT VECTORS: Data points that are closer to the hyperplane are called Support Vectors. To maximize the marginal distance between classifiers support vectors are used and if delete this support vector it will change the hyperplanes' position.

Naive Bayes

Naive Bayes is a supervised probabilistic machine learning algorithm that can be used for classification. Bayes Theorem Formula: Naive Bayes models are used recommendation systems, sentiment analysis, and spam filtering. Naive Bayes algorithms are very easy to implement. Types of Classifiers: Gaussian Naive Bayes

Bernoulli Naive Bayes

Multinomial Naive Bayes

Since our data is not discrete Gaussian Naïve Bayes approach is used.

3. ***Proposed method***

This section proposes the methodology and framework used for classification of comments. The steps involved are Normalization, standard Feature extraction, feature selection and finally classification.

Normalization: The Data set we have used contains list of comments and respective labels. These should be converted into feature vector which are used by our machine- learning algorithms. For this we use different Natural language processing techniques to obtain an accurate representation of the comments in feature vector form. We use various techniques based on our observations.

Removing unwanted strings: For the comments to be used by machine-learning algorithms they should be in standard form. Raw comments present in dataset which contains many unwanted strings like and many such encoding parts should be removed.

Hence the first step is to pre-process the comments by removing unwanted strings, hyphens and punctuations Correcting words: One of the reasons comments are classified as insulting is the presence of profane or abusive words. The total number of bad words present in comments is taken as one of the features. A dictionary of 500 bad words is compiled, which also includes variations of words (online forums sometimes use special characters to build an insulting word.

When we encounter such words, the dictionary helps to convert them into natural form. Also, Stemming is applied to capture bad word variations that are not contained in dictionary. Stemming reduces a word to its core root, for example embarrassing is reduced to embarrass. Here it is noted that stemming is only applied to bad word dictionary not on the dataset used, as it will lead to information loss.

Again, a small dictionary and a spell checker is used to convert all variations of "you"," you're" (e.g., u, ur etc) which are present in the dataset as participant use them as part of flexible language. Following Standard Feature Extraction: To train machine learning algorithms, strings should be converted in feature vector. We use frequency dictionary The process occurs in following steps and used it class as a pair for unique identification

Counting: Count the number of times each of these tokens occurs in a tweet along with it class it being added to a class which is 3X1 matrix.

Additional Features: Capturing pronouns: It is been observed that cyberaggressive comments which are directed towards peers are perceived more negatively and results in cyberbullying. Comments containing a pronoun like 'you' followed by an insulting or profane words are peer directed comments which are taken as negative and teens get frustrated after encountering such comments. So, to detect such comments we have used the count of pronouns as one of the features for detecting cyberbullying. To extract this feature, we calculate feature for pronoun present in comment. This feature is our strong hypothesis which greatly increases the accuracy and helps in detecting cyber-aggressive comments.

Feature Selection: The machine learning algorithms cannot handle all the features so we created whole new feature set consisting of three parameters positive count, negative count and bias term. These would reduce the time complexity and space complexity.

• Chi-Square Method: chi square (X2) method is commonly used for selecting best features. This metric calculates the cost of a feature using the value of the chi-squared statistics with respect to class. Classification: Once the features are built, we extract the best features using chi-squared test and apply the machine learning algorithms to train models on it. We have used SVM and logistic regression on our feature data. A brief summary of these algorithms is given below.

• Support vector machine (SVM): This algorithm maps the training data into feature space using kernel functions and then separates the dataset using large hyperplane. We have used non-linear kernel Rbf function

. • Logistic Regression: This algorithm provides probabilistic approach to data. The outcome are probabilities modelled as a function of predicted variables, using a logistic function.

5. Detection of cyberbullying incidents

Since effective prediction enables better targeted detection, we were interested in applying a similar methodology as in the prediction section to the training and testing of a detector. This distinction is of course the detection algorithm benefits from having access to the text comments from the discussion. In this section we only work on the data with non-zero negativity. The idea comes from having a first layer predictor with near to zero false positive.

In addition, a variety of non-text features were evaluated, including those features extracted from user behaviour (number of shared media objects, following, followers), media properties (likes, post time, caption) and image content. For example, we investigate the feature corresponding to the number of words. However, adding this feature does not provide any value to the classifier performance. It was observed that the number of words is considerably higher for examples of cyberbullying. The reason is the high correlation between the number of words and a set of variables with positive coefficients, namely “bitch”, “fuck”, “gay”, “hate”, “shut”, “suck”, “ugly”.

Similarly, we considered the “time interval” variable, i.e., the mean time between posts. This variable also has high correlation with cyberbullying indicator words and does not add to the classifier performance. Both of these supports our correlation analysis for “time interval” and “word count”. Another feature related to the media session is the number of likes the image has received, however it does not provide any improvement with a very small coefficient in the model. Another theme for future work is to obtain greater detail from the labelling surveys. Our experience was that streamlining the survey improved the response rate, quality and speed. However, we desire more detailed labelling, such as for different roles in cyberbullying identifying and differentiating the role of a victim’s defender, who may also spew negativity, from a victim’s bully or bullies. Finally, we can cascade our predictor with a more complicated detection algorithm to make examining cyberbullying-prone media sessions more scalable.

6. Discussion

While this paper has introduced prediction of cyberbullying in a media-based mobile social network, there remain a number of areas for improvement. One theme for future work is to improve the performance of our classifier and used it to various media related cyberbullying activities. New algorithms should be considered, such as RNN and LSTM. More input features should be evaluated, such as new image features, mobile sensor data, etc. Incorporating image features needs to be automated by applying image recognition algorithms. Temporal behaviour of comments for a posted media should be taken into account in designing the detection classifier. In this work we have only considered the image content and image and user metadata for prediction of cyberbullying. However, based on the improvement seen in using a small number of text comments, we think that considering the commenting history of users in previously shared media can prove to be useful.

Another theme for future work is to obtain greater detail from the labelling surveys. Our experience was that streamlining the survey improved the response rate, quality and speed. However, we desire more detailed labelling, such as for different roles in cyberbullying identifying and differentiating the role of a victim's defender, who may also spew negativity, from a victim's bully or bullies. Finally, we can cascade our predictor with a more complicated detection algorithm to make examining cyberbullying-prone media sessions more scalable.

7. Conclusion

An approach is proposed for detecting and preventing cyberbullying using Supervised Binary classification Machine Learning algorithms. Our model is evaluated on both Support Vector Machine and Naive Bayes, also for feature extraction, used the Frequency word dictionary. As the results show us that the accuracy for detecting cyberbullying content has also been great for Support Vector Machine non-linear of around 90.4% which is better than our [29]. Our model will help people from the attacks of social media bullies.

Acknowledgements

Acknowledgements and Reference heading should be left justified, bold, with the first letter capitalized but have no numbers. Text below continues as normal.

A. An example appendix

Authors including an appendix section should do so before References section. Multiple appendices should all have headings in the style used above. They will automatically be ordered A, B, C etc.

1. Example of a sub-heading within an appendix

There is also the option to include a subheading within the Appendix if you wish.

References

[1] Vimala Balakrishnan. Cyberbullying among young adults in malaysia: The roles of gender, age and internet frequency. Computers in Human Behavior, 46:149–157, 2015.

[2] Robin M Kowalski, Gary W Giumetti, Amber N Schroeder, and Heather H Reese. Cyber bullying among college students: Evidence from multiple domains of college life. In Misbehavior online in higher education. Emerald Group Publishing Limited, 2012.

[3] Sameer Hinduja and Justin W Patchin. Bullying beyond the schoolyard: Preventing and responding to cyberbullying. Corwin press, 2014.

[4] Michal Ptaszynski, Pawel Dybala, Tatsuaki Matsuba, Fumito Masui, Rafal Rzepka, and Kenji Araki. Machine learning and affect analysis against cyber-bullying. the 36th AISB, pages 7–16, 2010.

[5] Shane Murnion, William J Buchanan, Adrian Smales, and Gordon Russell. Machine learning and semantic analysis of in-game chat for cyberbullying. Computers & Security, 76:197–213, 2018.

[6] Maria Anzovino. Misogyny Detection on Social Media: A Methodological Approach. PhD thesis, Master's Thesis, Department of Informatics, Systems and Communication, 2018.

[7] Lu Cheng, Ruocheng Guo, and Huan Liu. Robust cyberbullying detection with causal interpretation. In Companion Proceedings of The 2019 World Wide Web Conference, pages 169–175, 2019.

[8] Simona Frenda, Somnath Banerjee, Paolo Rosso, and Viviana Patti. Do linguistic features help deep learning? the case of aggressiveness in mexican tweets. Computaci ́on y Sistemas, 24(2):633–643, 2020.

[9] Robin M Kowalski, Susan P Limber, and Annie McCord. A developmental approach to cyberbullying: Prevalence and protective factors. Aggression and Violent Behavior, 45:20–32, 2019.

[10] Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Emiliano De Cristofaro, Gianluca Stringhini, and Athena Vakali. Mean birds: Detecting aggression and bullying on twitter. In Proceedings of the 2017 ACM on web science conference, pages 13–22, 2017.

[11] Homa Hosseinmardi, Sabrina Arredondo Mattson, Rahat Ibn Rafiq, Richard Han, Qin Lv, and Shivakant Mishra. Detection of cyberbullying incidents on the instagram social network. arXiv preprint arXiv:1503.03909, 2015.

[12] Mei Sze Choo. Cyberbullying on Facebook and psychosocial adjustment in Malaysian adolescents. PhD thesis, [Honolulu]:[University of Hawaii at Manoa],[December 2016], 2016.

[13] Robert R McCrae and Oliver P John. An introduction to the five-factor model and its applications. Journal of personality, 60(2):175–215, 1992.

[14] Oliver P John, Sanjay Srivastava, et al. The big five trait taxonomy: History, measurement, and theoretical perspectives. Handbook of personality: Theory and research, 2(1999):102–138, 1999.

[15] Qianyun Jiang, Fengqing Zhao, Xiaochun Xie, Xingchao Wang, Jia Nie, Li Lei, and Pengcheng Wang. Difficulties in emotion regulation and cyberbullying among chinese adolescents: a mediation model of loneliness and depression. Journal of interpersonal violence, 37(1-2):NP1105– NP1124, 2022.

[16] Qianjia Huang, Vivek Kumar Singh, and Pradeep Kumar Atrey. Cyber bullying detection using social and textual analysis. In Proceedings of the 3rd International Workshop on Socially-aware Multimedia, pages 3–6, 2014.

[17] Nazanin Alavi, Taras Reshetukha, Eric Prost, Kristen Antoniak, Charmy Patel, Saad Sajid, and Dianne Groll. Relationship between bullying and suicidal behaviour in youth presenting to the emergency department. Journal of the Canadian

Academy of Child and Adolescent Psychiatry, 26(2):70, 2017.

[18] A Saravanaraj, JI Sheeba, and S Pradeep Devaneyan. Automatic detection of cyberbullying from twitter. IRACST-International J. Comput. Sci. Inf. Technol. Secur, 6(2016):2249–9555, 2016.

[19] Karthik Dinakar, Roi Reichart, and Henry Lieberman. Modeling the detection of textual cyberbullying. In Proceedings of the International AAAI Conference on Web and Social Media, volume 5, pages 11–17, 2011.

[20] Steven L Salzberg. C4. 5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993, 1994.

[21] Ying Chen, Yilu Zhou, Sencun Zhu, and Heng Xu. Detecting offensive language in social media to protect adolescent online safety. In 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing, pages 71–80. IEEE, 2012.

[22] Maral Dadvar, Rudolf Berend Trieschnigg, and Franciska MG de Jong. Expert knowledge for automatic detection of bullies in social networks. In 25th Benelux Conference on Artificial Intelligence, BNAIC 2013, pages 57–64. Delft University of Technology, 2013.

[23] Maral Dadvar, FMG de Jong, Roeland Ordelman, and Dolf Trieschnigg. Improved cyberbullying detection using gender information. In Proceedings of the Twelfth Dutch-Belgian Information Retrieval Workshop (DIR 2012). University of Ghent, 2012.

[24] Maral Dadvar, Dolf Trieschnigg, Roeland Ordelman, and Franciska de Jong. Improving cyberbullying detection with user context. In European Conference on Information Retrieval, pages 693–696. Springer, 2013.

[25] Claudia Peersman, Walter Daelemans, and Leona Van Vaerenbergh. Predicting age and gender in online social networks. In Proceedings of the 3rd international workshop on Search and mining usergenerated contents, pages 37–44, 2011.

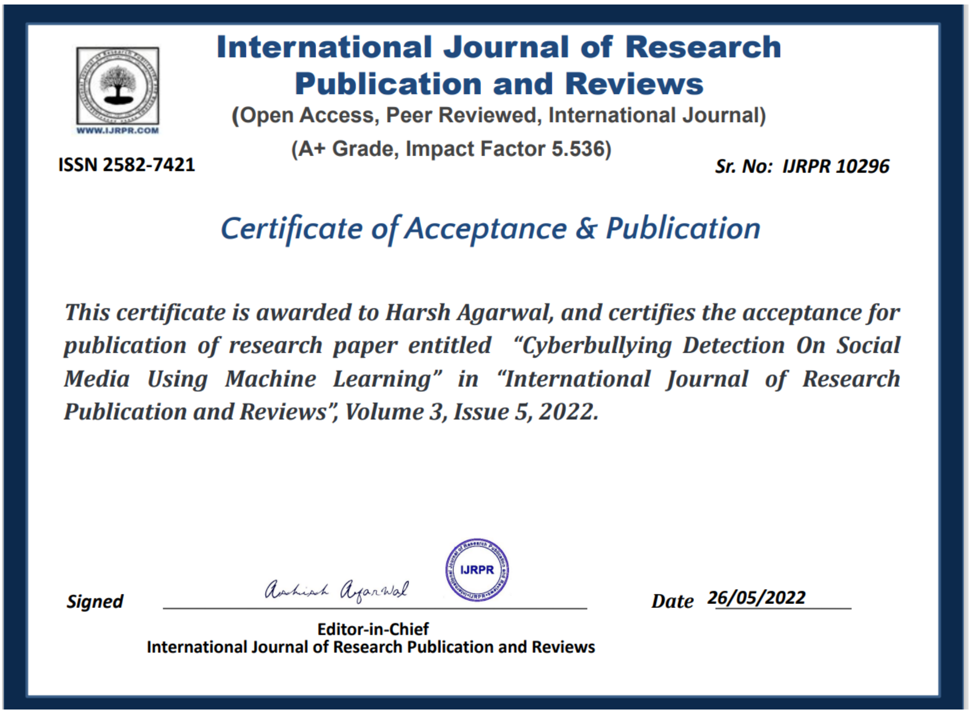
[26] Anna Squicciarini, Sarah Rajtmajer, Y Liu, and Christopher Griffin. Identification and characterization of cyberbullying dynamics in an online social network. In Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015, pages 280–285, 2015.

[27] K Nalini and L Sheela. Classification of tweets using text classifier to detect cyber bullying. In Emerging ICT for Bridging the Future-Proceedings of the 49th Annual Convention of the Computer Society of India CSI Volume 2, pages 637–645. Springer, 2015.

[28] Vikas S Chavan and SS Shylaja. Machine learning approach for detection of cyber-aggressive comments by peers on social media network. In 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pages 2354–2358. IEEE, 2015.

[29] Rahul Ramesh Dalvi, Sudhanshu Baliram Chavan, and Aparna Halbe. Detecting a twitter cyberbullying using machine learning. In 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), pages 297–301. IEEE, 2020

B.2 CERTIFICATES



WWW.IJRPR.COM

International Journal of Research Publication and Reviews

(Open Access, Peer Reviewed, International Journal)

(A+ Grade, Impact Factor 5.536)

ISSN 2582-7421

Sr. No: IJRPR 10296

Certificate of Acceptance & Publication

This certificate is awarded to Harsh Agarwal, and certifies the acceptance for publication of research paper entitled "Cyberbullying Detection On Social Media Using Machine Learning" in "International Journal of Research Publication and Reviews", Volume 3, Issue 5, 2022.

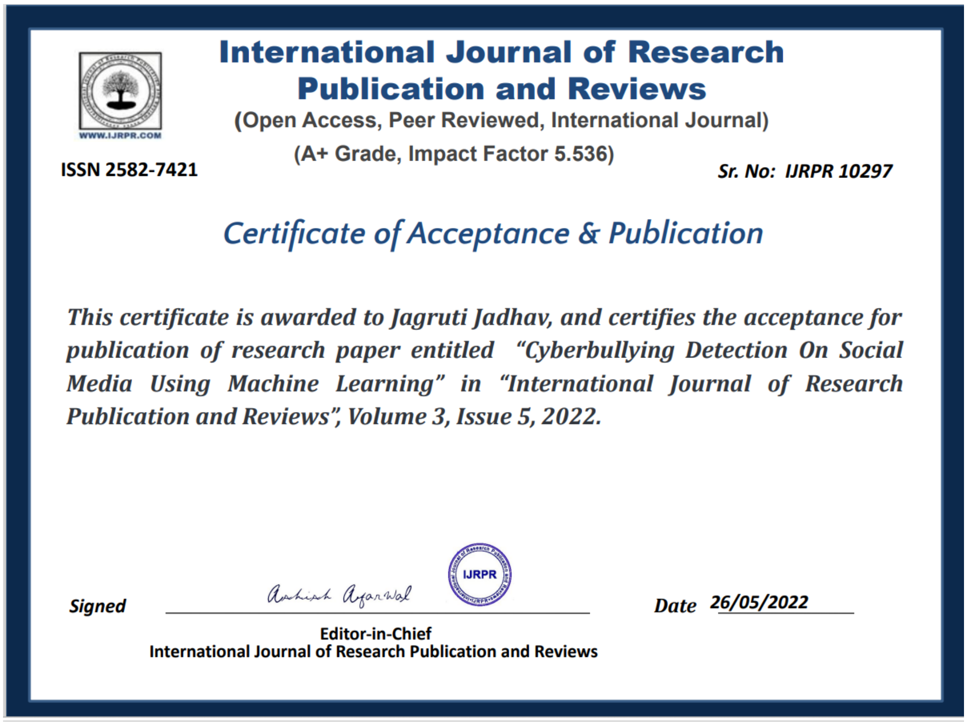
Signed Ashish Agarwal

International Journal of Research Publication and Reviews • IJRPR •

IJRPR

Date ***26/05/2022***

Editor-in-Chief
International Journal of Research Publication and Reviews



WWW.IJRPR.COM

International Journal of Research Publication and Reviews

(Open Access, Peer Reviewed, International Journal)

(A+ Grade, Impact Factor 5.536)

ISSN 2582-7421

Sr. No: IJRPR 10297

Certificate of Acceptance & Publication

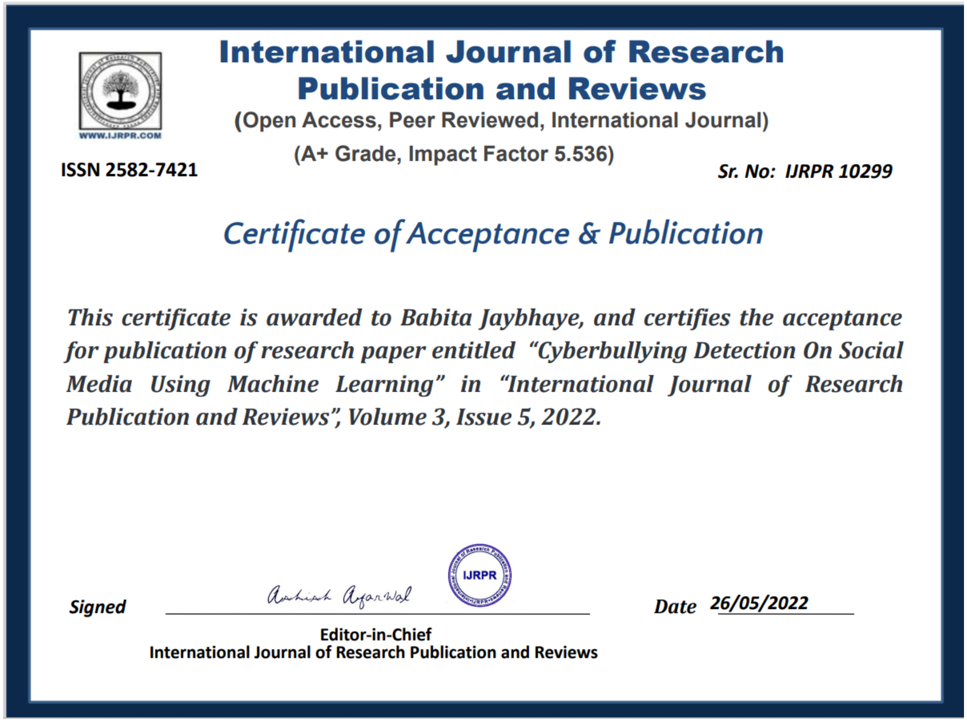
This certificate is awarded to Jagruti Jadhav, and certifies the acceptance for publication of research paper entitled "Cyberbullying Detection On Social Media Using Machine Learning" in "International Journal of Research Publication and Reviews", Volume 3, Issue 5, 2022.

IJRPR

Signed Ashish Agarwal

Editor-in-Chief
International Journal of Research Publication and Reviews

Date 26/05/2022



WWW.IJRPR.COM

International Journal of Research Publication and Reviews

(Open Access, Peer Reviewed, International Journal)

(A+ Grade, Impact Factor 5.536)

ISSN 2582-7421

Sr. No: IJRPR 10299

Certificate of Acceptance & Publication

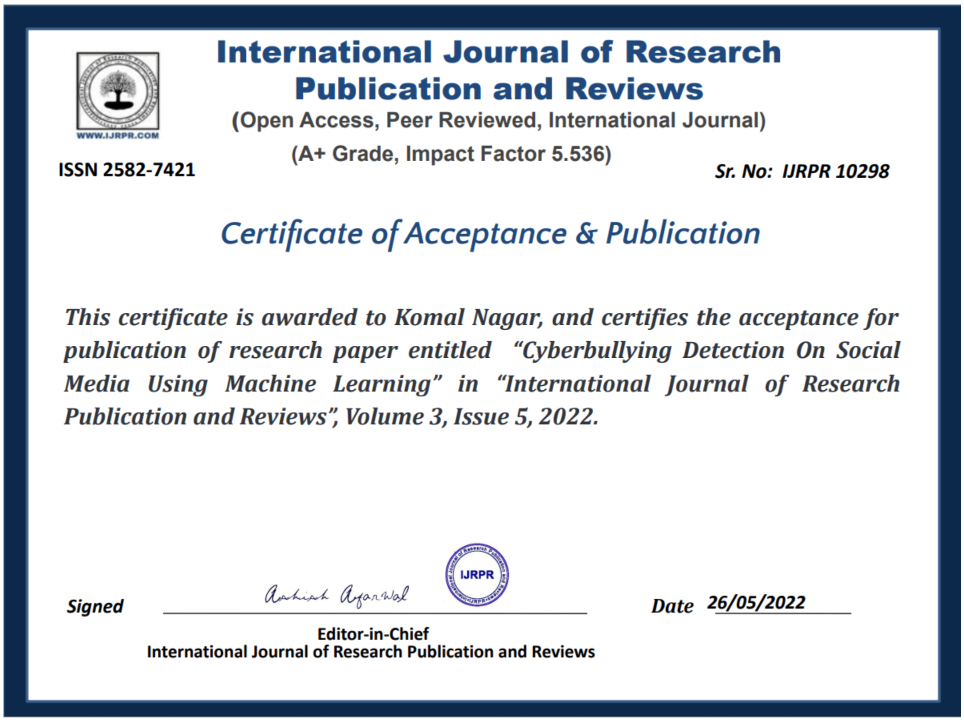
This certificate is awarded to Babita Jaybhaye, and certifies the acceptance for publication of research paper entitled "Cyberbullying Detection On Social Media Using Machine Learning" in "International Journal of Research Publication and Reviews", Volume 3, Issue 5, 2022.

IJRPR

Signed Ashish Agarwal

Editor-in-Chief
International Journal of Research Publication and Reviews

Date 26/05/2022



WWW.IJRPR.COM

International Journal of Research Publication and Reviews

(Open Access, Peer Reviewed, International Journal)

(A+ Grade, Impact Factor 5.536)

ISSN 2582-7421

Sr. No: IJRPR 10298

Certificate of Acceptance & Publication

This certificate is awarded to Komal Nagar, and certifies the acceptance for publication of research paper entitled "Cyberbullying Detection On Social Media Using Machine Learning" in "International Journal of Research Publication and Reviews", Volume 3, Issue 5, 2022.

Signed Aashish Agarwal

IJRPR

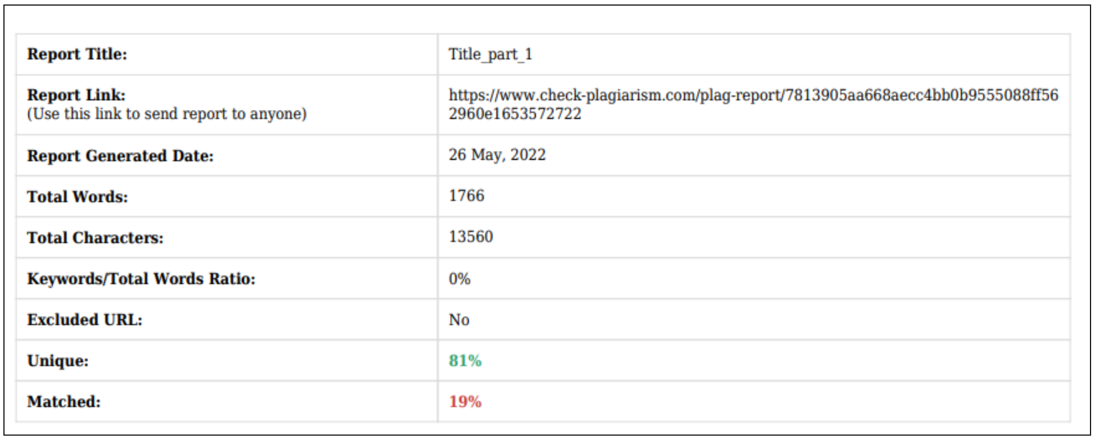
Editor-in-Chief
International Journal of Research Publication and Reviews

Date **26/05/2022**

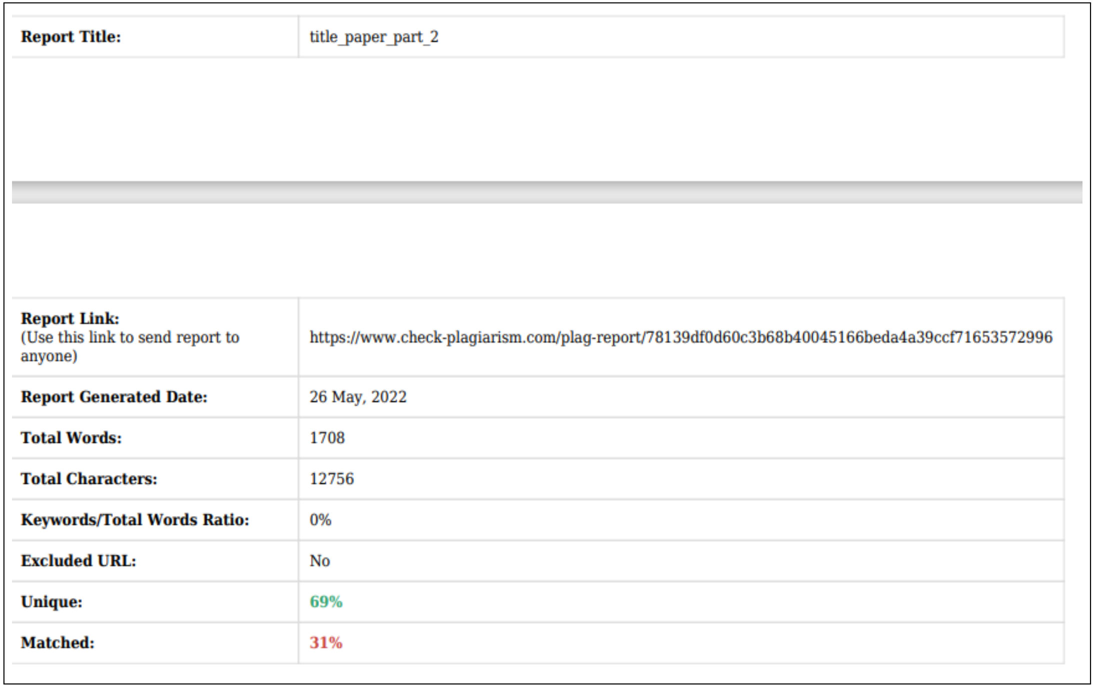
ANNEXURE C

PLAGIARISM REPORT

Plagiarism report



Report Title:	Title_part_1
Report Link: (Use this link to send report to anyone)	https://www.check-plagiarism.com/plag-report/7813905aa668aecc4bb0b9555088ff562960e1653572722
Report Generated Date:	26 May, 2022
Total Words:	1766
Total Characters:	13560
Keywords/Total Words Ratio:	0%
Excluded URL:	No
Unique:	**81%**
Matched:	**19%**



Report Title:	title_paper_part_2
Report Link: (Use this link to send report to anyone)	https://www.check-plagiarism.com/plag-report/78139df0d60c3b68b40045166beda4a39ccf71653572996
Report Generated Date:	26 May, 2022
Total Words:	1708
Total Characters:	12756
Keywords/Total Words Ratio:	0%
Excluded URL:	No
Unique:	**69%**
Matched:	**31%**