

A REPORT ON

Cyberbullying Detection in Social Media Using Supervised ML & NLP Techniques

**SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE IN
THE PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE
DEGREE**

OF

BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)

SUBMITTED BY

Gayatri Battise

EXAM NO: B150364233

Jivan Devhare

EXAM NO: B150364282

Anklesh Patil

EXAM NO: B150364430

Anuj Patil

EXAM NO: B150364431



Sinhgad Institutes

DEPARTMENT OF COMPUTER ENGINEERING

**STES's SMT KASHIBAI NAVALE COLLEGE OF ENGINEERING
VADGAON BK, OFF SINHGAD ROAD, PUNE 411041**

SAVITRIBAI PHULE PUNE UNIVERSITY 2021 - 22



Sinhgad Institutes

CERTIFICATE

This is to certify that the project report entitles

Cyberbullying Detection in Social Media Using Supervised ML & NLP Techniques

Submitted by

Gayatri Battise

EXAM NO: B150364233

Jivan Devhare

EXAM NO: B150364282

Anklesh Patil

EXAM NO: B150364430

Anuj Patil

EXAM NO: B150364431

is a bonafide work carried out by them under the supervision of Dr.K.N.Honwadkar and it is approved for the partial fulfilment of the requirement of Savitribai Phule Pune University, for the award of the degree of Bachelor of Engineering (Computer Engineering)
This project work has not been earlier submitted to any other Institute or University for the award of any degree.

Dr.K.N.Honwadkar

Dr.R.H.Borhade

Guide/Co-Guide

Head

Department of Computer Engineering

Department of Computer Engineering

Dr. A.V.Deshpande

Principal

Place: Pune

Date:

ACKNOWLEDGMENT

It gives us great pleasure in presenting the preliminary project report on **Cyberbullying Detection in Social Media Using Supervised ML & NLP Techniques**. With due respect and gratitude we would like to take this opportunity to thank our internal guide **Dr.K.N.Honwadkar** for giving us all the help and guidance we needed. We are really grateful for his kind support. He has always encouraged us and given us the motivation to move ahead. He has put in a lot of time and effort in this project along with us and given us a lot of confidence. We are also grateful to **Dr.R.H.Borhade** Head of Computer Engineering Department, VIIT College of Engineering for his indispensable support. Also we wish to thank all the other people who have helped us in the successful completion of this project. We would also like to extend our sincere thanks to Principal Dr.A.V.Deshpande, for his dynamic and valuable guidance the project and providing the necessary facilities that helped us to complete our dissertation work. We would like to thank my colleagues friends who have helped us directly or indirectly to complete this work.

Anklesh Patil
Anuj Patil
Gayatri Battise
Jivan Devhare

.

ABSTRACT

From the day internet came into existence, the era of social networking sprouted. In the beginning, no one may have thought the internet would be a host of numerous amazing services the social networking. Today we can say that online applications and social networking websites have become a non-separable part of one's life. Many people from diverse age groups spend hours daily on such websites. DespEven thoughtlet are emotionally connected together through media, these facilities bring along big threats with them such as cyber-attacks, which includes include lying. As social networking sites are increasing, cyberbullying is increasing day by day. To identify word similarities in the tweets made by bullies and make use of machine learning and can develop an ML model automatically detects social media bullying actions. However, many social media bullying detection techniques have been implemented, but many of them were textual-based. Under this background and motivation, it can help to prevent the happen of cyberbullying if we can develop relevant techniques to discover cyberbullying in social media. A machine learning model is proposed to detect and prevent bullying on Twitter. Naïve Bayes is used for training and testing social media bullying content.

Contents

List of Figures

List of Tables

INTRODUCTION.....	1
1.1 Overview	1
1.2 Motivation.....	1
1.3 Problem Definition and Objectives	2
1.3.1 Problem Definition.....	2
1.3.2 Objectives.....	2
1.4 Project Scope and Limitation	2
1.4.1 Project Scope	2
1.4.2 Limitation	2
1.5 Methodologies of Problem solving.....	3
LITERATURE REVIEW	4
SOFTWARE REQUIREMENTS SPECIFICATION.....	6
3.1 Assumptions and Dependencies	6
3.2 Functional Requirements	6
3.2.1 System Feature	6
3.3 External Interface Requirements	6
3.3.1 User Interfaces.....	6
3.3.2 Hardware Interfaces ..	6
3.3.3 Software Interfaces.....	6
3.3.4 Communication Interfaces.....	7
3.4 Nonfunctional Requirements.....	7
3.4.1 Performance Requirements.....	7
3.4.2 Safety Requirements.....	7
3.4.3 Security Requirements.....	7
3.4.4 Software Quality Attributes.....	7
3.5 System Requirements	8
3.5.1 Database Requirements.....	8

3.5.2	Software Requirements(Platform Choice).....	8
3.5.3	Hardware Requirements.....	8
3.6	Analysis Models: SDLC Model to be applied.....	9
SYSTEM DESIGN		11
4.1	System Architecture.....	11
4.2	Mathematical Model.....	12
4.3	Data Flow Diagrams:	13
4.3.1	Level 0 data flow diagram.....	13
4.4	UML Diagrams.....	16
4.4.1	Use case Diagram.....	16
4.4.2	Class Diagram :.....	17
4.4.3	Activity Diagram:.....	18
4.4.4	Component Diagram.....	20
4.4.5	Deployment Diagram.....	21
Project Plan.....		22
5.1	Project Estimates	22
5.1.1	Reconciled Estimates	22
5.1.2	Project Resources.....	22
5.2	Risk Management	22
5.2.1	Risk Identification.....	22
5.2.2	Risk Analysis.....	23
5.3	Project Schedule	24
5.3.1	Project task set.....	24
5.3.2	Task network.....	24
5.3.3	Timeline Chart.....	25
5.4	Team Organization.....	25
5.4.1	Team structure.....	25
5.4.2	Management reporting and communication.....	26
Project Implementation.....		27
6.1	Overview of Project Modules.....	27
6.2	Tools and Technologies Used.....	27

6.2.1 Technology Description.....	27
6.2.2 Hardware Specifications	28
6.2.3 Software Specifications.....	28
Software Testing	29
7.1 Types Of Testings	29
7.1.1 Unit testing.....	29
7.1.2 Integration testing.....	29
7.1.3 Functional test	29
7.1.4 System Test.....	30
7.1.5 White Box Testing	30
7.1.6 Black Box Testing.....	30
7.1.7 Unit Testing:.....	30
7.1.8 Integration Testing.....	31
7.1.9 Acceptance Testing	31
7.2 Test cases and Test Results	31
TEST CASE NO. 02.....	32
TEST CASE NO. 04	34
TEST CASE NO. 05	35
Results.....	37
8.1 Outcomes	37
8.2 Screen Shots	38
CONCLUSIONS.....	48
9.1 Conclusions	48
9.2 Future Work	48
9.3 REFERENCES	49

List of Figures

4.1	System Architecture	12
4.2	Level 0 Data Flow Diagram	14
4.3	Level 1 Data Flow Diagram	15
4.4	Level 2 Data Flow Diagram	16
4.5	USE CASE	17
4.6	Class Diagram	18
4.7	Activity Diagram	20
4.8	Component Diagram	21
4.9	Deployment Diagram	22
5.1	Task network	25
5.2	Timeline Chart	26
5.3	Timeline Chart	27
5.4	Team structure	27

List of Tables

5.1	Risk Analysis	24
5.2	Management reporting and communication	28
7.1	Test Cases	35
7.2	Test Cases	36

Chapter 1

INTRODUCTION

1.1 Overview

Online social networking sites like Twitter, Facebook, Instagram and some online social networking companies have become extremely popular in recent years. People spend a lot of time in OSN making friends with people they are familiar with or interested in. Twitter, founded in 2006, has become one of the most popular microblogging service sites. Around 200 million users create around 400 million new tweets a day for spam growth. Twitter spam, known as unsolicited tweets containing malicious links that the non-stop victims to external sites containing the spread of malware, spreading malicious links, etc., hit not only more legitimate users, but also the whole platform. Consider the example because during the election of the Australian Prime Minister in 2013, a notice confirming that his Twitter account had been hacked. Many of his followers have received direct spam messages containing malicious links. The ability to order useful information is essential for the academic and industrial world to discover hidden ideas and predict trends on Twitter. However, spam generates a lot of noise on Twitter. To detect spam automatically, researchers applied machine learning algorithms to make spam detection a classification problem. Ordering a tweet broadcast instead of a Twitter user as spam or non-spam is more realistic in the real world.

1.2 Motivation

- System reports the impact of the data related factors, such as spam to nonspam ratio, training data size, and data sampling, to the detection performance.
- System extracts 12 lightweight features for streaming tweet spam detection
- System creates a big ground-truth for the research on spam tweet detection.
- System investigates machine learning algorithms to build up the tweet spam detection model.

1.3 Problem Definition and Objectives

1.3.1 Problem Definition

Proposed system design problem for tweet analysis and tweet classification for generating summary for tweet event. To generate summary system uses statement using tweet words are considered. Proposed system used to broadcast tweet summary on social media. In addition to that we formulate problem for short URL spamming detection

1.3.2 Objectives

- To classify spam tweets and not spam.
- To classify tweets based on tags and tweets based on links.
- To improve the performance and accuracy of the result.

1.4 Project Scope and Limitation

1.4.1 Project Scope

- Use of this system is in online social networking for spam detection.
- Feature of spam tweets seems to be time varying.
- It is unable to detect the categorization of tweets on the basis of their types.

1.4.2 Limitation

Not Worked on Shared link.

1.5 Methodologies of Problem solving

Algorithm

Given training dataset D which consists of documents belonging to different class say Class A and Class B

Calculate the prior probability of class A = number of objects of class A / total number of objects

Calculate the prior probability of class B = number of objects of class B / total number of objects

Find N_i , the total no of frequency of each class

N_a = the total no of frequency of class A

N_b = the total no of frequency of class B

Find conditional probability of keyword occurrence given a class:

$P(\text{value 1/Class A}) = \text{count} / n_i(A)$

$P(\text{value 1/Class B}) = \text{count} / n_i(B)$

$P(\text{value 2/Class A}) = \text{count} / n_i(A)$

$P(\text{value 2/Class B}) = \text{count} / n_i(B)$

..

..

..

$P(\text{value } n/\text{Class B}) = \text{count} / n_i(B)$

Avoid zero frequency problems by applying uniform distribution

Classify Document C based on the probability $p(C/W)$

Find $P(A/W) = P(A) * P(\text{value 1/Class A}) * P(\text{value 2/Class A}) * \dots * P(\text{value } n/\text{Class A})$

Find $P(B/W) = P(B) * P(\text{value 1/Class B}) * P(\text{value 2/Class B}) * \dots * P(\text{value } n/\text{Class B})$

Assign document to class that has higher probability.

Chapter 2

LITERATURE REVIEW

- F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida [1] in this paper, we consider the problem of detecting spammers on Twitter. First we collect a large set of Twitter data which includes over 54 million users, 1.9 billion connections and almost 1.8 billion tweets. Use of tweets related to three famous trend themes of 2009, we build a great label Collection of users, manually classified in spammers and not spammer. So we identify a series of features Related to the content of the tweet and the social behavior of the user that could potentially be used to detect spammers. We He used these features as machine learning attributes process to classify users as spammers or nonspammers. Our strategy manages to detect a large part of the spammers while only a small percentage of non-spammers they are incorrectly classified about 70 of spammers and 96.Spammers have not been classified correctly. Also our results highlight the most important attributes for spam detection on Twitter.
- G. Biau [2] Random forest is a scheme proposed by Leo Breiman in the 2000s to build a predictor Set with a series of decision trees that grow in subspaces of randomly selected data. Despite growing interest and practical use, there has been little exploration of the statistical properties of Random forests and little is known about the mathematical forces that drive the algorithm. In this paper, we offer an in-depth analysis of a random forest model suggested by Breiman (2004), which is very close to the original algorithm. We show in particular that the procedure is consistent and it adapts itself to scarcity, in the sense that its rate of convergence depends only on the number of forts characteristics and not on the number of noise variables present.
- M. Bishop [3] the work reported in this summary is concentrated. Mainly on the significant theoretical and experimental. Developments that have advanced the state of the art Automatic model recognition and automatic learning. The automatic recognition of the models is identification and Assignment of pattern classes for machines. The models presented for identification can be visual, Oral or electromagnetic. It is usually done what the heart of the recognition of more realistic models. The problems are the use of "typical patterns" or "learning"

Observations "to determine the decision procedure. Used by the car; Thus, the study of the automatic. The recognition of the model inevitably involves the study of automatic learning. • Chen, J. Zhang, X. Chen, Y. Xiang, and W. Zhou [4] twitter has changed the mode of communication and Receive news for people's daily lives in recent years. Meanwhile, Due to the popularity of Twitter, it also becomes a main objective. For spamming activities. To stop spammers, Twitter is using Google Safe Browsing to detect and block spam links. Although blacklists can block malicious URLs embedded tweets, its delay in time hinders the ability to protect users in real time so, researchers start to apply different machines learning algorithms to detect spam from Twitter. However, there is no comprehensive evaluation of the performance of each algorithm. To detect spam from Twitter in real time due to the lack of large size the fundamental truth. To carry out a thorough evaluation, we have collected a large data set of over 600 million public tweets. More labeled about 6.5 million tweets of spam and 12 light extract Features that can be used for online tracking. In Furthermore, we conducted a series of experiments in six machine learning algorithms in various conditions to improve Understands its effectiveness and weakness for timely Twitter Spam detection we will make our data set labeled for researchers. Those interested in validating or expanding our work. • M. Egele, G. Stringhini, C. Kruegel, and G. Vigna [5] in this paper, we present a new approach to detect User accounts compromised in social networks and we apply at two popular social networking sites, Twitter and Facebook. Our approach uses a composition of statistics. Detection of patterns and anomalies to identify accounts Experience a sudden change in behavior. From the behavior Changes may also be due to benign reasons (for example, a user it could change your favorite client application or post updates in an unusual moment), it is necessary to derive a mode of Distinguish between harmful and legitimate changes. Thats why we look for groups of accounts that everyone experiences. Similar changes in a short period of time, assuming that these changes are the result of a harmful campaign. This is developing. We have developed an instrument, called COMPA, that implements our approach and we execute it on a large scale data set of over 1.4 billion Twitter publicly available messages, as well as in a data set of 106 million Facebook COMPA messages may identify compromised accounts in both.

SOFTWARE REQUIREMENTS SPECIFICATION

3.1 Assumptions and Dependencies

- The user should have knowledge about the attributes set assigned.
- The User should have basic design in mind.
- The system is developed in Java platform.

3.2 Functional Requirements

3.2.1 System Feature

This system used for spam detection

3.3 External Interface Requirements

3.3.1 User Interfaces

Home page User and Admin Login page Upload File Page Check duplication Page. Analysis Graph Page.

3.3.2 Hardware Interfaces

The entire software requires a completely equipped computer system including monitor, keyboard, and other input output devices.

3.3.3 Software Interfaces

The system can use Microsoft as the operating system platform. System also makes use of certain GUI tools. To run this application we need JDK 1.8 and above as java platform and Apache tomcat as server. To store data we need MySQL database.

3.3.4 Communication Interfaces

Communication using Java and servlet APIs

3.4 Nonfunctional Requirements

3.4.1 Performance Requirements

The performance of the system lies in the way it is handled. Every user must be given proper guidance regarding how to use the system. The other factor which affects the performance is the absence of any of the suggested requirements.

3.4.2 Safety Requirements

To ensure the safety of the system, perform regular monitoring of the system so as to trace the proper working of the system. An authenticated user is only able to access system.

3.4.3 Security Requirements

Any unauthorized user should be prevented from accessing the system. Password authentication can be introduced.

3.4.4 Software Quality Attributes

Accuracy: -

The level of accuracy in the proposed system will be higher. All operation would be done correctly and it ensures that whatever information is coming from the center is accurate. Result is organic results.

Reliability: -

The reliability of the proposed system will be high due to the above stated reasons.

The reason for the increased reliability of the system is that now there would be proper storage of information and tweet analysis model.

3.5 System Requirements

3.5.1 Database Requirements

MySQL Database

3.5.2 Software Requirements(Platform Choice)

Operating System - Windows 7 and above

Application Server - Apache Tomcat

Front End - HTML, JDK 1.7, JSP

Scripts - JavaScript.

Server side Script - Java Server Pages.

Database - My SQL 5.0

IDE - Eclipse

Framework - Apache Spark

3.5.3 Hardware Requirements

Processor - Pentium 4

Speed - 1.1 GHz

RAM - 256 MB(min)

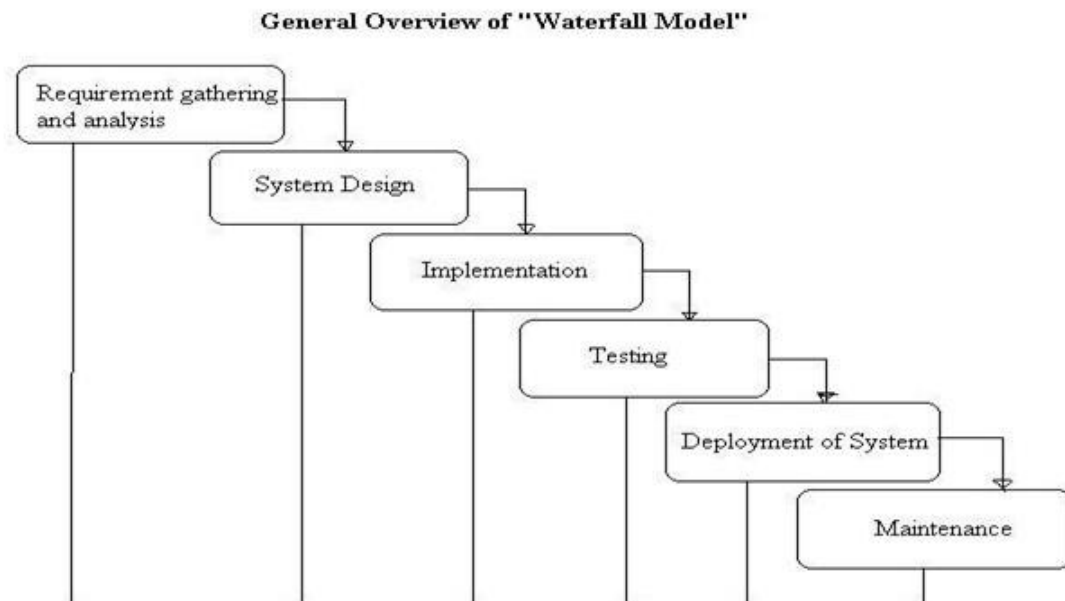
Hard Disk - 20 GB

Key Board - Standard Windows Keyboard

Mouse - Two or Three Button Mouse

Monitor - SVGA

3.6 Analysis Models: SDLC Model to be applied



The Waterfall Model is sequential design process, often used in Software development processes, where progress is seen as flowing steadily down through the phase of conception, Initiation, Analysis, Design, Construction, Testing, Production/Implementation and Maintenance. This Model is also called as the classic Life cycle model as it suggests a systematic sequential approach to software developments. This one of the oldest model followed in software engineering. The process begins with the communication phase where the customer specifies the requirements and then progress through other phases like planning, modeling, construction and deployment of the software.

There are 5 Phase of water fall model:

The Waterfall Model is sequential design process, often used in Software development processes, where progress is seen as flowing steadily down through the phase of conception, Initiation, Analysis, Design, Construction, Testing, Production/Implementation and Maintenance. This Model is also called as the classic Life cycle model as it suggests a systematic sequential approach to software developments. This one of the oldest model followed in software engineering. The process begins with the communication phase where the customer specifies the requirements and then progress through other phases like planning, modeling, construction and deployment of the software.

There are 5 Phase of water fall model:

1. COMMUNICATION

In communication phase the major task performed is requirement gathering which helps in finding out exact need of customer. Once all the needs of the customer are gathered the next step is planning.

2. PLANNING

In planning major activities like planning for schedule, keeping tracks on the processes and the estimation related to the project are done. Planning is even used to find the types of risks involved throughout the projects. Planning describes how technical tasks are going to take place and what resources are needed and how to use them.

3. MODELING

This is one the important phases as the architecture of the system is designed in this phase. Analysis is carried out and depending on the analysis a software model is designed. Different models for developing software are created depending on the requirements gathered in the first phase and the planning done in the second phase.

4. CONSTRUCTION

The actual coding of the software is done in this phase. This coding is done on the basis of the model designed in the modeling phase. So in this phase software is actually developed and tested.

5. DEPLOYMENT

In this last phase the product is actually rolled out or delivered installed at customers end and support is given if required. A feedback is taken from the customer to ensure the quality of the product. From the last two decades Waterfall model has come under lot of criticism due to its efficiency issues. So lets discuss the advantages and disadvantages of waterfall model.

SYSTEM DESIGN

4.1 System Architecture

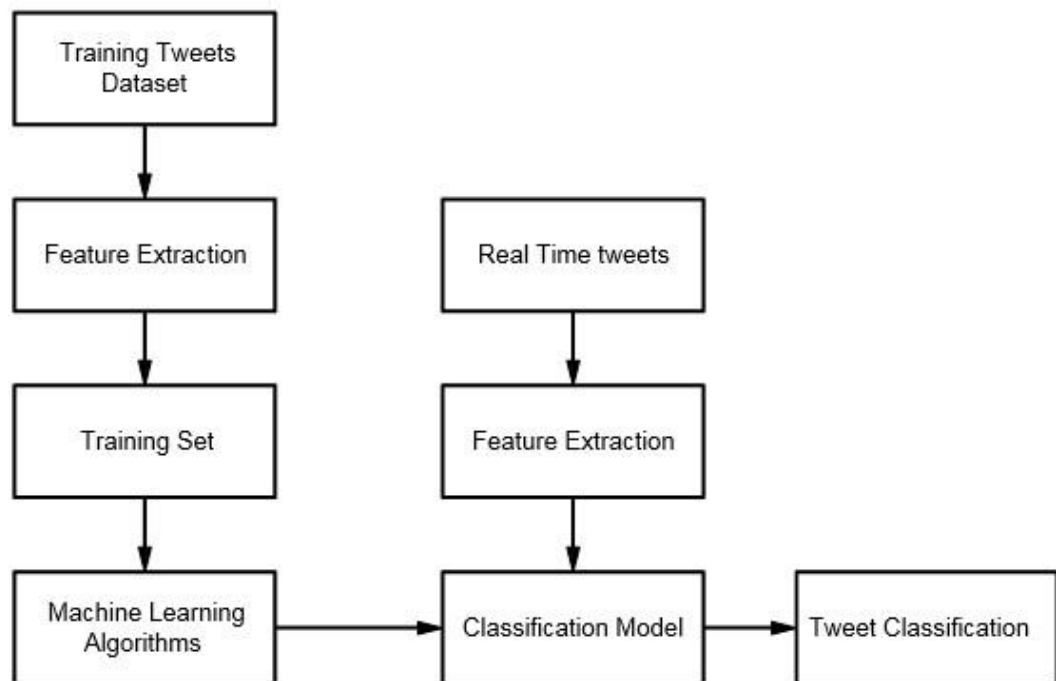


Figure 4.1: System Architecture

4.2 Mathematical Model

Let us consider S as a system for Twitter Spam Detection.

S=

INPUT:

Identify the inputs

$F = f_1, f_2, f_3, \dots, f_N$ F as set of functions to execute commands.

$I = i_1, i_2, i_3$ sets of inputs to the function set

$O = o_1, o_2, o_3$ Set of outputs from the function sets,

$S = I, F, O$

$I = \text{Tweet}$

$O = \text{Output i.e. Spam or Non-spam}$ $F =$

Functions implemented to get the output

Space Complexity:

The space complexity depends on Presentation and visualization of discovered patterns.

More the storage of data more is the space complexity.

Time Complexity:

Check No. of patterns available in the datasets= n

If (n1) then retrieving of information can be time consuming. So the time complexity of this algorithm is $O(n^2)$.

= Failures and Success conditions.

Failures:

1. Huge database can lead to more time consumption to get the information.
2. Hardware failure.
3. Software failure.

Success:

1. Search the required information from available in Datasets.
2. User gets result very fast according to their needs.

4.3 Data Flow Diagrams:

4.3.1 Level 0 data flow diagram

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.

2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.

3. Figure 4.1 shows level 0 DFD which shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.

Level 0 Data Flow Diagram

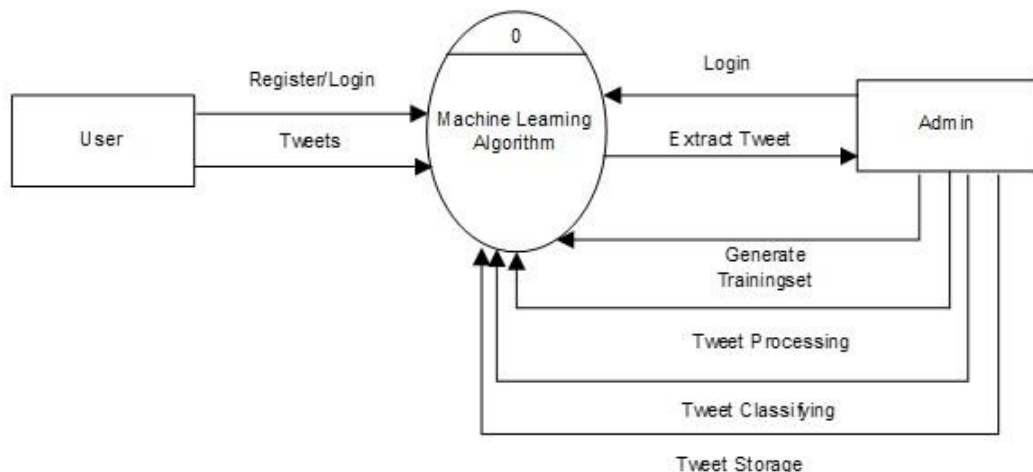


Figure 4.2: Level 0 Data Flow Diagram

Level 1 Data Flow Diagram

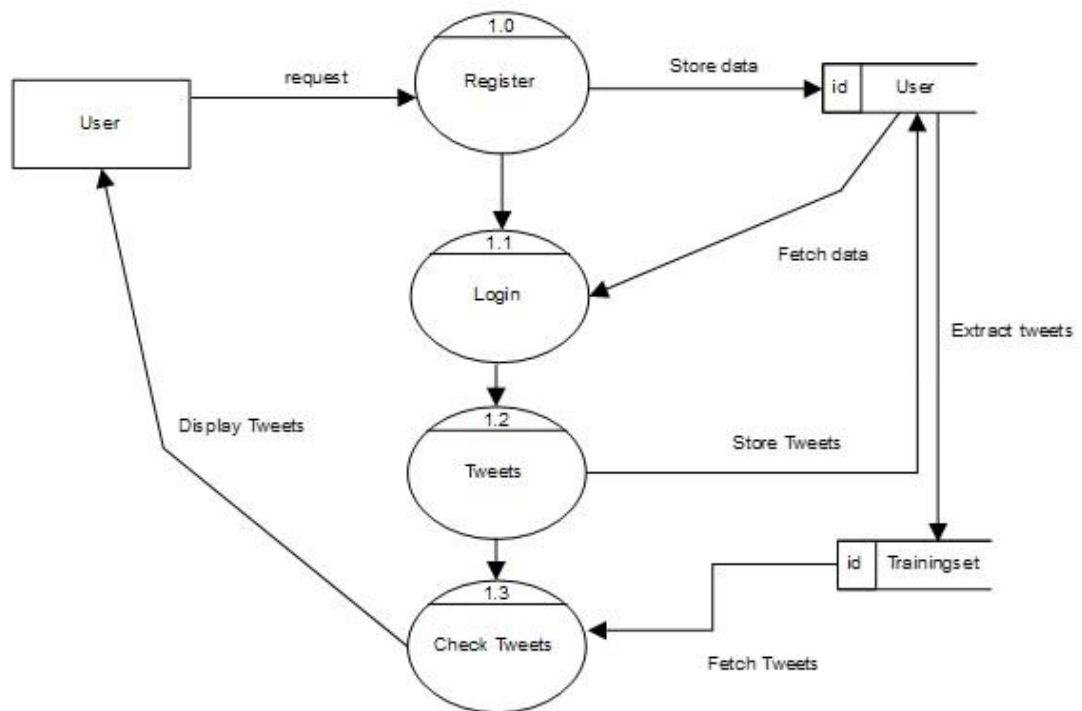


Figure 4.3: Level 1 Data Flow Diagram

Level 2 Data Flow Diagram

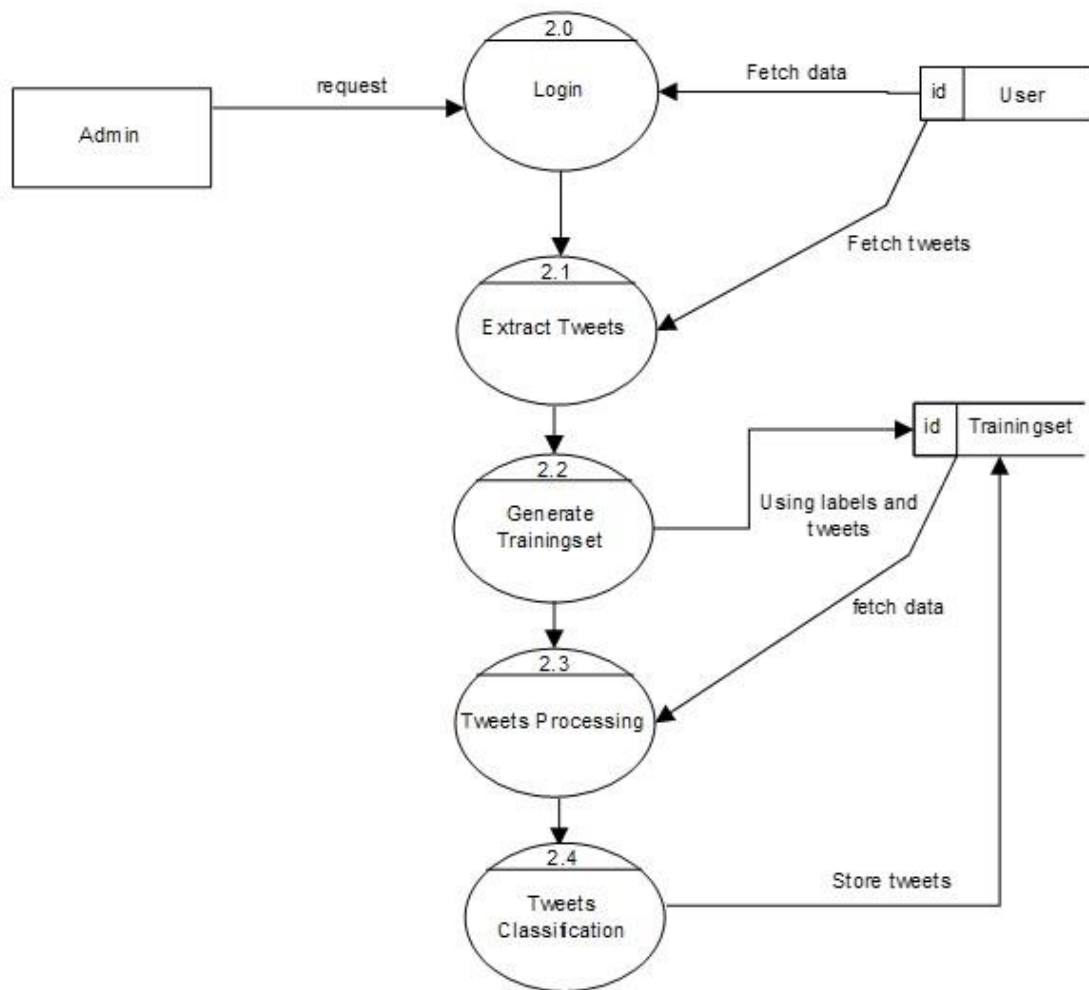


Figure 4.4: Level 2 Data Flow Diagram

4.4 UML Diagrams

4.4.1 Use case Diagram

A use case diagram is a graphical representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can show the different types of users of a system and the various ways in which they interact with the system. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analyzed to gather its functionality use cases are prepared and actors are identified. The purposes of use case diagrams can be as follows:

Used to gather requirements of a system.

Used to get an outside view of a system.

Identify external and internal factors influencing the system.

Show the interaction among the actors.

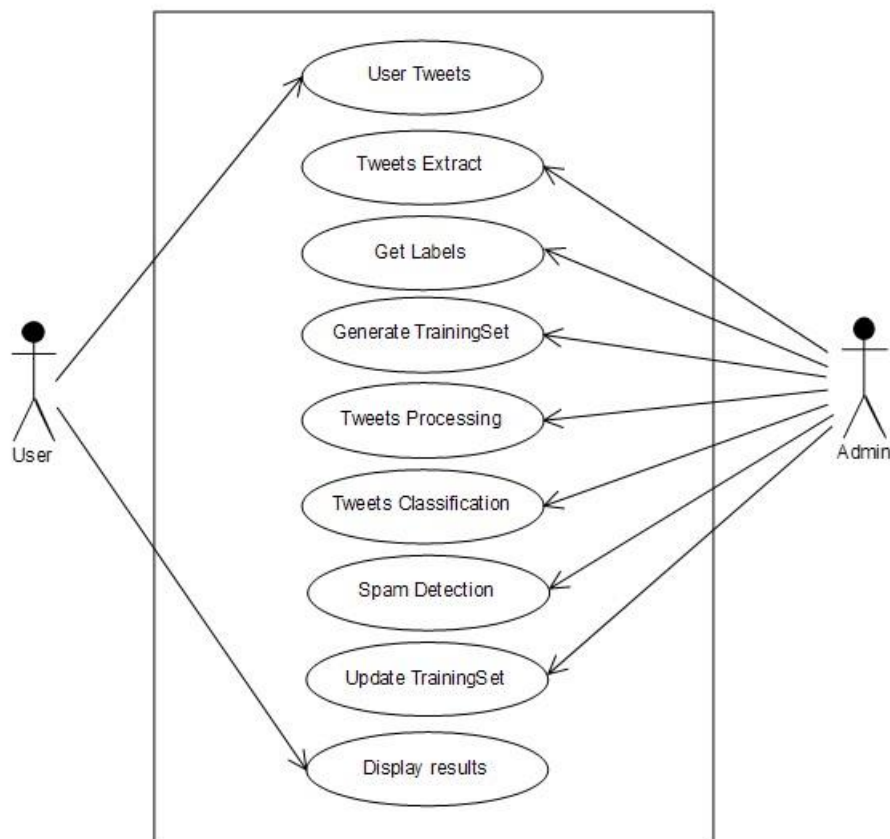


Figure 4.5: USE CASE

4.4.2 Class Diagram :

The class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application. The class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams which can be mapped directly with object oriented languages. The class diagram shows a collection of classes, interfaces, associations, collaborations and constraints. It is also known as a structural diagram. The purpose of the class diagram is to model the static view of an application.

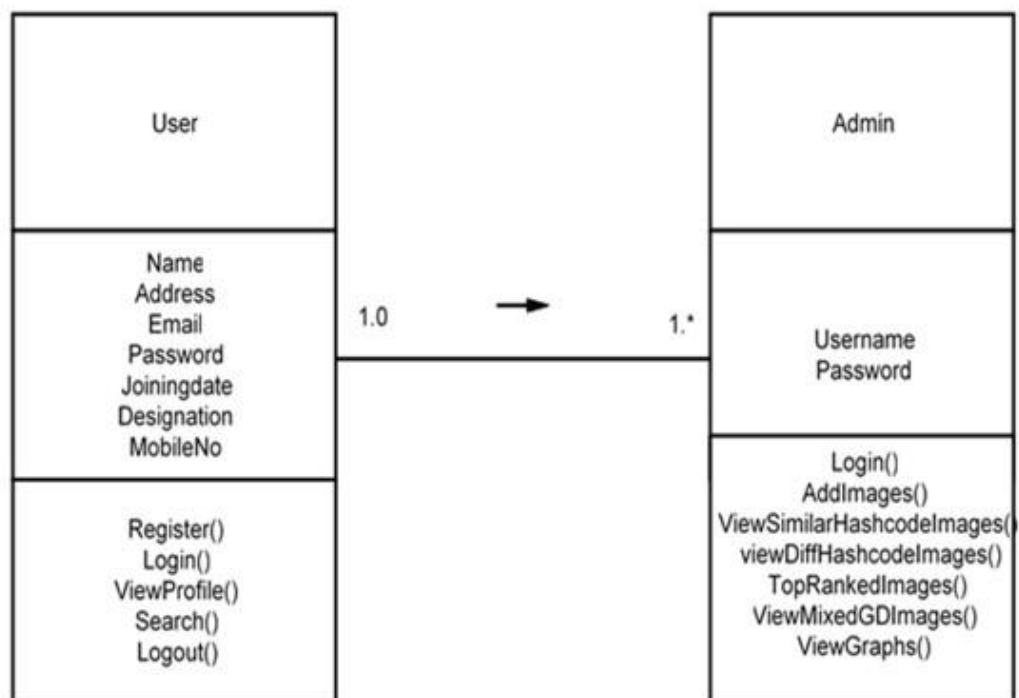


Figure 4.6: Class Diagram

4.4.3 Activity Diagram:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows). Activity diagrams show the overall flow of control. Activity diagrams are constructed from a limited number of shapes, connected with arrows. The most important shape types:

Rounded rectangles represent actions;

Diamonds represent decisions;

Bars represent the start (split) or end (join) of concurrent activities; A black circle represents the start (initial state) of the workflow;

An encircled black circle represents the end (final state).

Arrows run from the start towards the end and represent the order in which activities happen. Hence they can be regarded as a form of flowchart. Typical flowchart techniques lack constructs for expressing concurrency. However, the join and split symbols in activity diagrams only resolve this for simple cases; the meaning of the model is not clear when they are arbitrarily combined with decisions or loops.

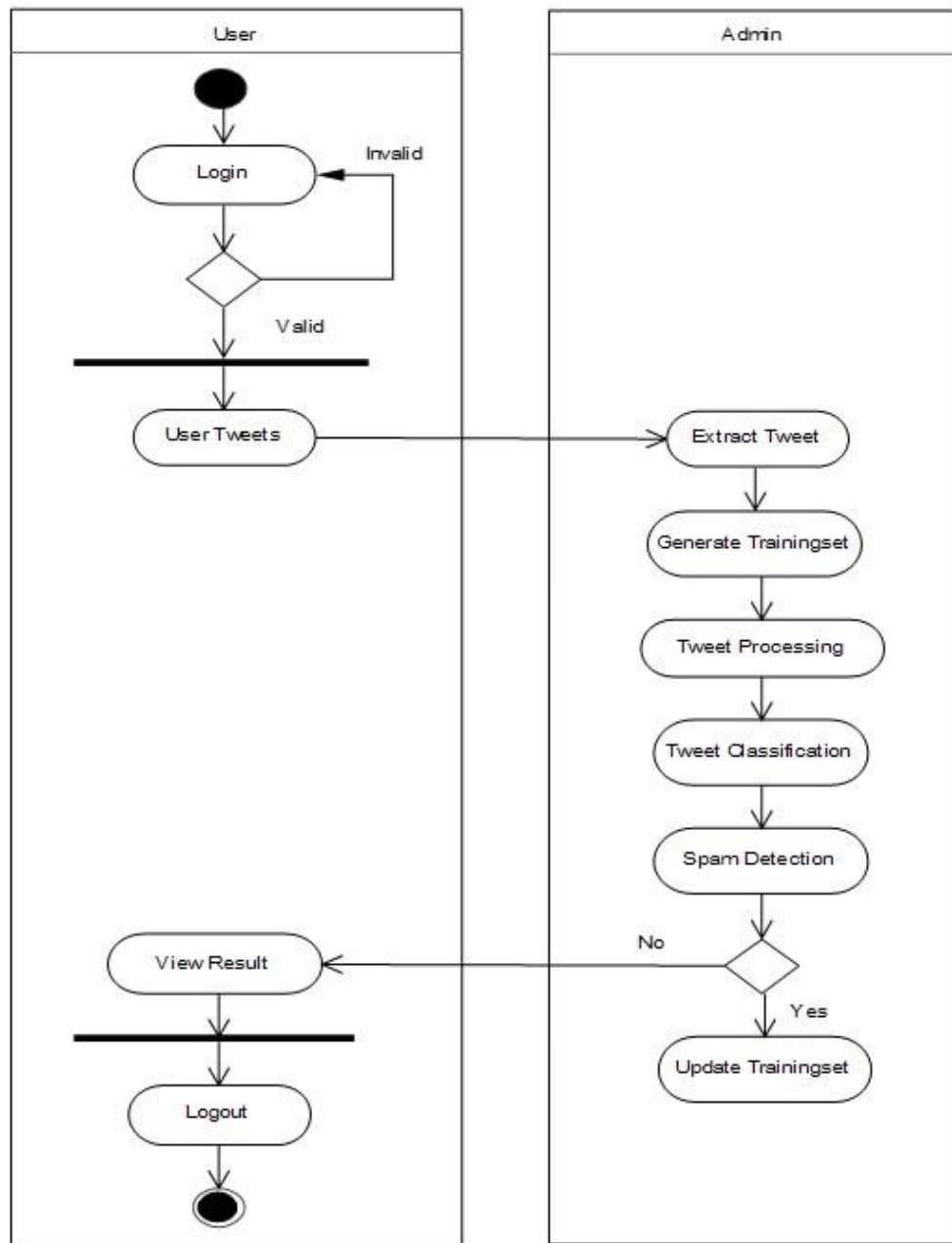


Figure 4.7: Activity Diagram

4.4.4 Component Diagram

A Component Diagram displays the structural relationship of components of a software system. These are mostly used when working with complex systems that have many components. Components communicate with each other using interfaces. The interfaces are linked using connectors.

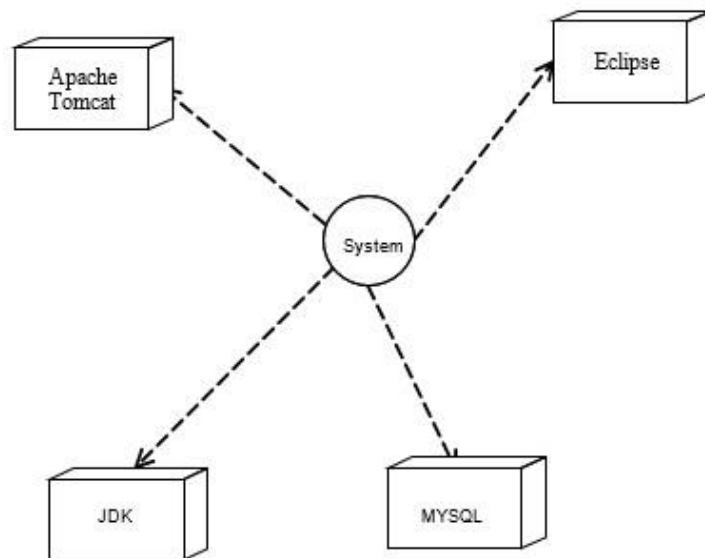


Figure 4.8: Component Diagram

4.4.5 Deployment Diagram

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed. So deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.

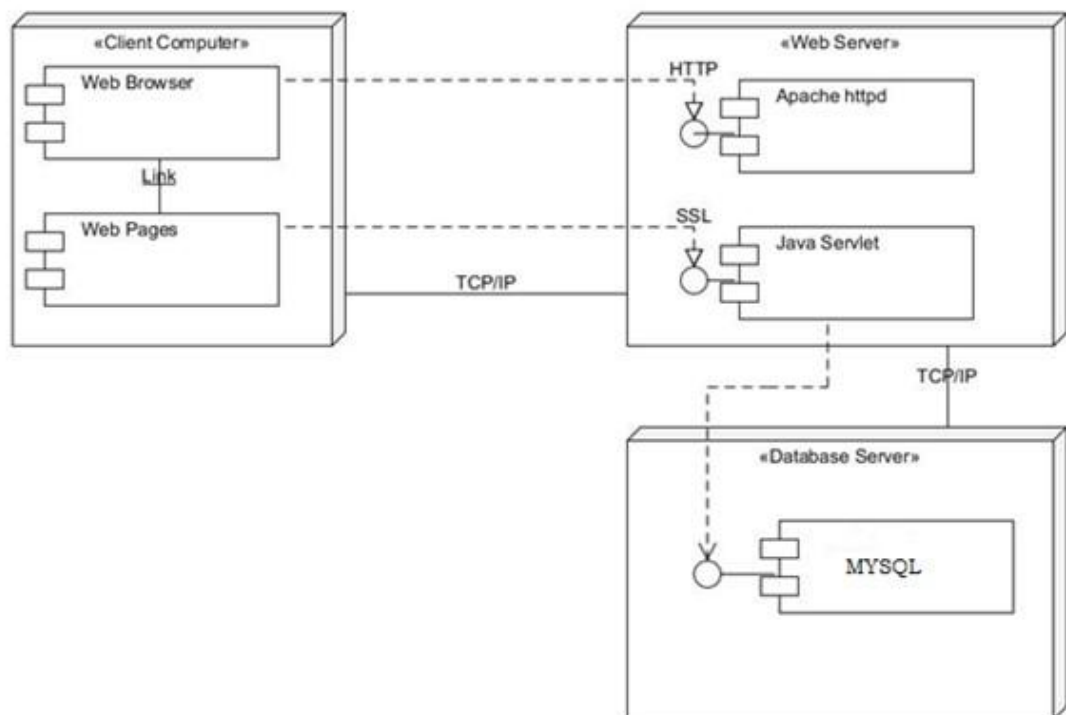


Figure 4.9: Deployment Diagram

Chapter 5

Project Plan

5.1 Project Estimates

5.1.1 Reconciled Estimates

Cost Estimate

Cost will estimate after completing the project that depend on time to complete the project. Also efforts required to complete.

Time Estimates

Time will depend on modules of project. Also project plan of execution.

5.1.2 Project Resources

1. Hardware Resources Required

System: Pentium IV 2.4 GHz. Hard Disk: 40 GB. Floppy Drive: 44 Mb.
Monitor: 15 VGA Color.

2. Software Resources Required

Operating system: Windows. Coding Language: Java 1.8 Database: MySql 5
IDE: Eclipse

5.2 Risk Management

5.2.1 Risk Identification

For risks identification, review of scope document, requirements specifications and schedule is done. Answers to questionnaire revealed some risks. Each risk is categorized as per the categories mentioned in [?]. Please refer table ?? for all the risks. You can refereed following risk identification questionnaire.

1. Have top software and customer managers formally committed to support the project?

Answer: Yes , Have top software and customer managers formally committed to support the project

2. Are end-users enthusiastically committed to the project and the system/product to be built?

Yes, end-users enthusiastically committed to the project and the system/product to be built

3. Are requirements fully understood by the software engineering team and its customers?

Yes, Are requirements fully understood by the software engineering team and its customers

5. Have customers been involved fully in the definition of requirements?

Yes, customers been involved fully in the definition of requirements

6. Are project requirements stable?

Answer: all project requirements are stable

9. Is the number of people on the project team adequate to do the job? Yes, the number of people on the project team adequate to do the job

5.2.2 Risk Analysis

DESCRIPTION	LOW	High
Login detail	no	yes
Internet connection slow	yes	no

Table 5.1: Risk Analysis

Following are the details for each risk.

Risk ID 1

Risk Description Description 1

1. While login to system validation is there so user must follow rules and proper entries

Risk ID 2

1. While registering to system internet connection should be there for registration to get users data online

5.3 Project Schedule

5.3.1 Project task set

Major Tasks in the Project stages are:

- Task 1: Requirement Analysis (Base Paper Explanation).
- Task 2: Project Specification (Paper Work).
- Task 3: Technology Study and Design.
- Task 4: Coding and Implementation (Module Development).

5.3.2 Task network

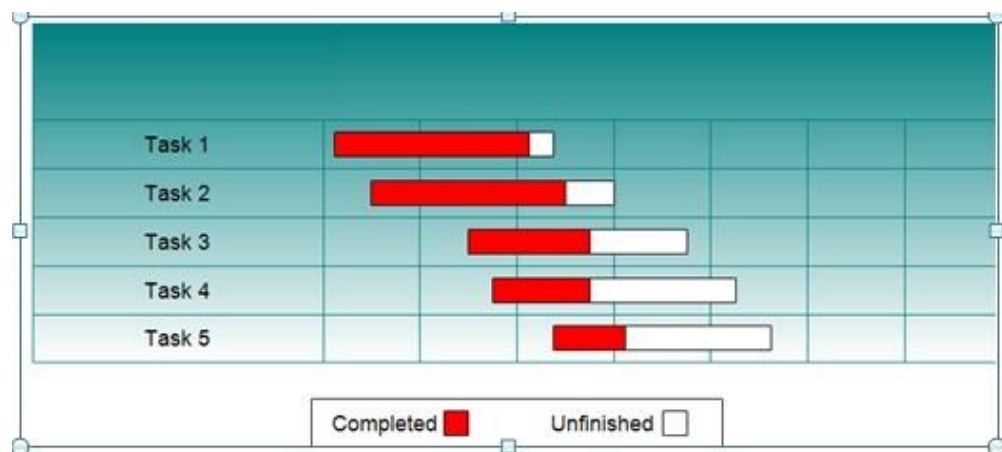


Figure 5.1: Task network

5.3.3 Timeline Chart

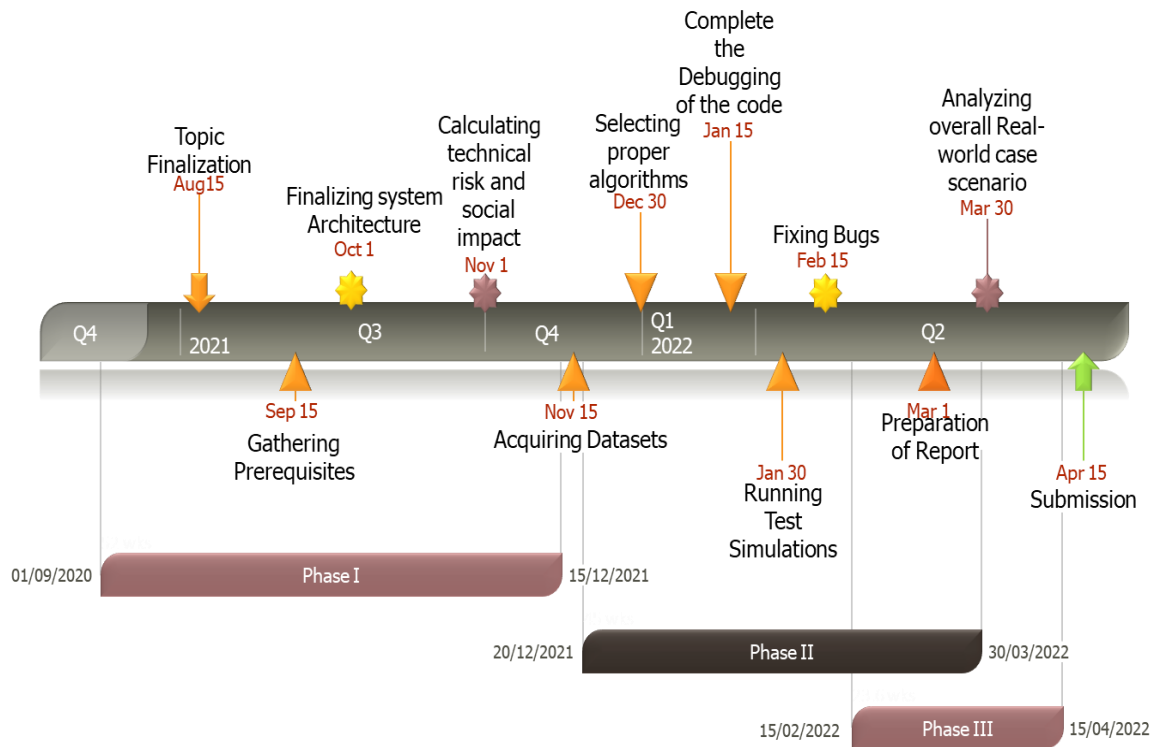


Figure 5.2: Timeline Chart

5.4 Team Organization

5.4.1 Team structure

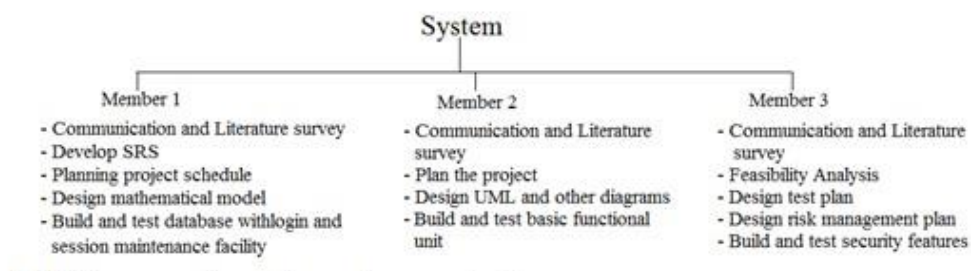



Figure 5.4: Team structure

5.4.2 Management reporting and communication



Sr No.	Month	Description
1	June	Discussion with guide regarding domain. Searching for IEEE paper for domain.
2	July	Short listing of IEEE papers within domain. Selection of IEEE paper.
3	August	Deciding Project name. Submission of Synopsis.
4	September	Requirement analysis. Designing of models.
5	October	Report preparation. Stage-I report submission.




Table 5.2: Management reporting and communication

Chapter 6

Project Implementation

6.1 Overview of Project Modules

1. User Module

The user can be defined as an individual accessing the application. User has the features of posting

any comment or status in the wall or tweet anything. i.

Account authentication

Through this feature, a user is authenticated into the system. The login credentials of the user must match the information stored in the application. On successful provisioning of the credentials, a user is logged in the system ii. Posting a Tweet

The user of the application has the right to send or post a sort of text lines in the form of tweet in the application.

2. Admin Module

The admin has the complete right over the system. The admin can view List of Valid and Invalid i.e. spam post. i. Add training tweet

This Module will facilitate Admin to specify sample tweet content containing valid or spam tweet. This email will be further analyzed to store some training parameters as valid or spam tweet.

6.2 Tools and Technologies Used

6.2.1 Technology Description

In the Java programming language, all source code is first written in plain text files ending with the .java extension. Those source files are then compiled into

.class files by the javac compiler. A .class file does not contain code that is native to your processor; it instead contains bytecodes — the machine language of the Java Virtual Machine¹ (Java VM). The java launcher tool then runs your application with an instance of the Java Virtual Machine.

6.2.2 Hardware Specifications

Processor - I3,I5
Speed - 3.8 GHz
RAM - 4GB
Hard Disk - 1 TB
Key Board - Standard Windows Keyboard
Mouse - Two or Three Button Mouse
Monitor - LCD(Liquid Crystal Display)

6.2.3 Software Specifications

Operating System: Windows 10
Programming Language: JAVA
Backend: Mysql 5.0
IDE : Eclipse Oxygen
Tool:Apache Framework

Chapter 7

Software Testing

7.1 Types Of Testings

7.1.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

7.1.2 Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

7.1.3 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

7.1.4 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

7.1.5 White Box Testing

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose.

It is purpose. It is used to test areas that cannot be reached from a black box level.

7.1.6 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot see into it. The test provides inputs and responds to outputs without considering how the software works.

7.1.7 Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

7.1.8 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or one step up software applications at the company level interact without error.

7.1.9 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements. Test Results: All the test cases mentioned above passed successfully.

No defects encountered.

7.2 Test cases and Test Results

TEST CASE NO. 01

Aim:

Analysis of comments on Mr. Donald Triumph's tweets and find the negative and positive comments. So that we can check whether that parent tweet is getting positive or negative feedback from the public to identify the cyberbullying case.

Dataset: Dataset imported from Kaggle.com

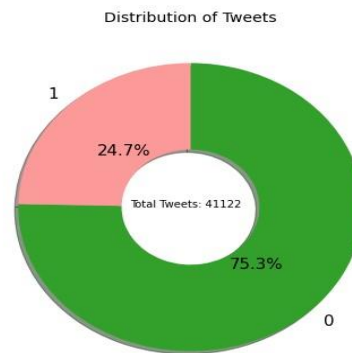
Output:

Total No. of Tweets	Violent Tweets (in %)	Non-Violent Tweets
41122	24.7%	75.3%

Result:

- Out of the total of 41122 tweets done by the public on the Mr. Donal Triumphs tweet **24.7%** are Violent(positive) tweets and **75.3%** are nonviolent.
- This shows us there is no major activity of cyberbullying in this test case 01.
- We are getting an accuracy 85%.

Graph:



TEST CASE NO. 02

Aim:

Analysis of comments on NFT tweets and find the negative and positive comments. So that we can check whether that parent tweet is getting positive or negative feedback from the public to identify the cyberbullying case.

Dataset: Dataset imported from Kaggle.com

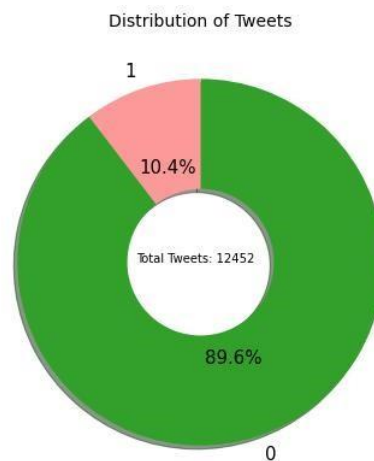
Output:

Total No. of Tweets	Violent Tweets (in %)	Non-Violent Tweets
12452	10.1%	89.6%

Result:

- Out of the total of 12452 tweets done by the public on the NFT tweet 10.1% are Violent(positive) tweets and 89.6% are non-violent.
- This shows us there is no major activity of cyberbullying in this test case 01.
- We are getting an accuracy 94%.

Graph:



TEST CASE NO. 04

Aim:

Analysis of comments on Demonetization tweets and find the negative and positive comments. So that we can check whether that parent tweet is getting positive or negative feedback from the public to identify the cyberbullying case.

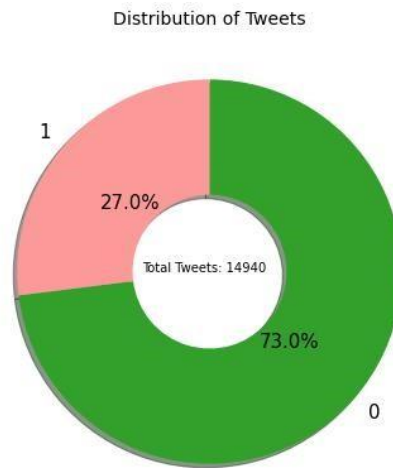
Dataset: Dataset imported from
Kaggle.com Output:

Total No. of Tweets	Violent Tweets (in %)	Non-Violent Tweets
14940	27.0%	73.0%

Result:

- Out of the total of 14940 tweets done by the public on the Demonetization tweets 27.0% are Violent(positive) tweets and 73.0% are non-violent.
- This shows us there are significant violent tweets in this category.
- We are getting an accuracy 92%.

Graph:



TEST CASE NO. 05

Aim:

Analysis of comments on Stock-market tweets and find the negative and positive comments. So that we can check whether that parent tweet is getting positive or negative feedback from the public to identify the cyberbullying case.

Dataset: Dataset imported from Kaggle.com **Output:**

Total No. of Tweets	Violent Tweets (in %)	Non-Violent Tweets
583	2.1%	97.9%

Result:

- Out of the total of 14940 tweets done by the public on the stock-market tweet 2.1% are Violent(positive) tweets and 97.9% are non-violent.
- This shows us there are significant non-violent tweets in this category.
- We are getting an accuracy 99%.

Graph:

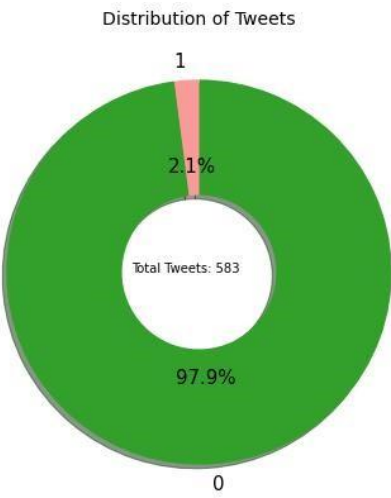


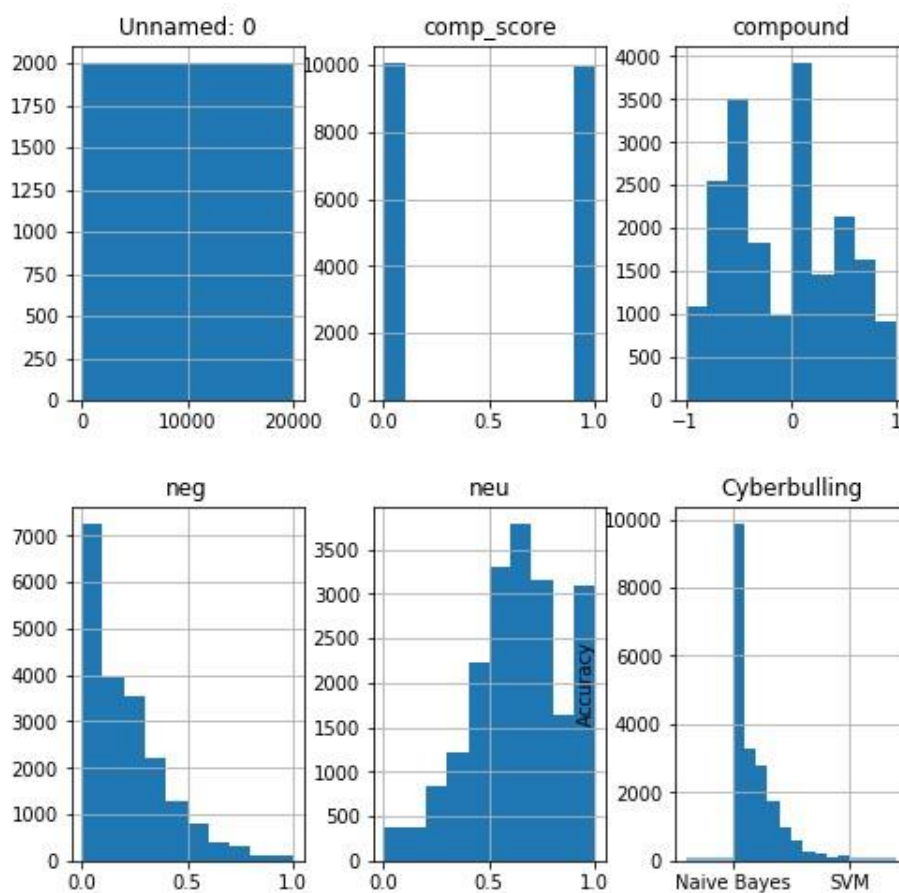
Table 7.2: Test Cases

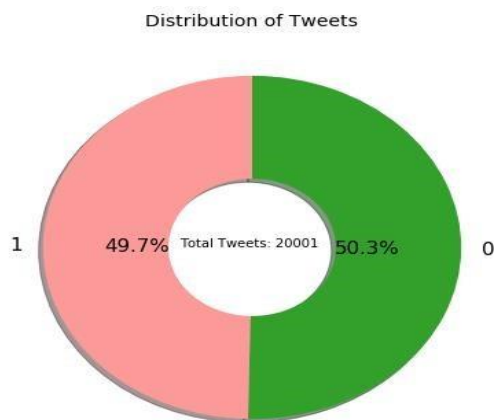
Chapter 8

Results

8.1 Outcomes

Add result graph screenshot here.....





8.2 Screen Shots

Add all project screenshots here??????????

Sentiment De... — □ ×

Input your Tweet

C. Pawan: Sentiment

Tweet was rated as:

Tweet was rated as:

Tweet was rated as:

Tweet Overall Rated As:

Clear

Exit

Sentiment De... — □ ×

Input your Tweet

Good morning

Check Sentiment

Tweet was rated as:

0.0% Negative

Tweet was rated as:

25.6% Neutral

Tweet was rated as:

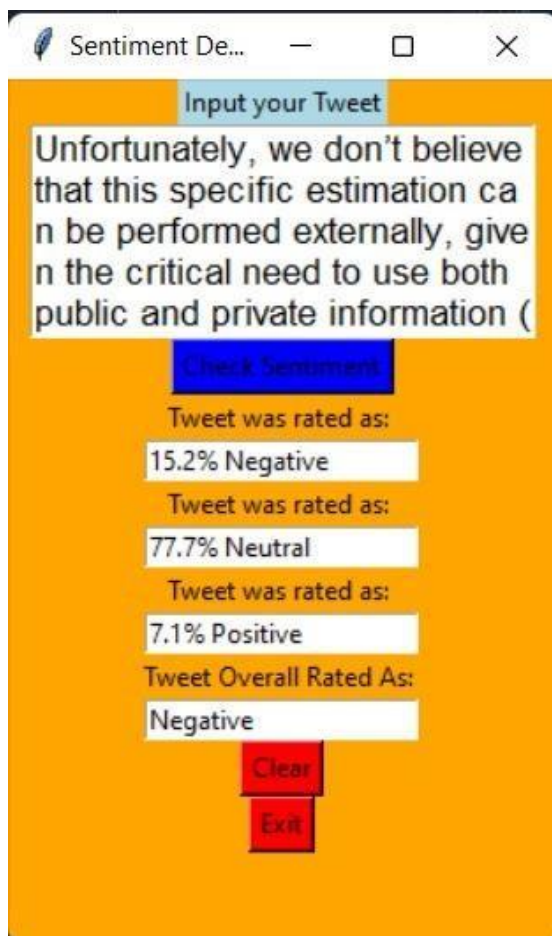
74.4% Positive

Tweet Overall Rated As:

Positive

Clear

Exit



Index	content	user	compound	neg	ne
0	Get fucking real dude.	scotthamilton	0	0	1
1	She is as dirty as the...	mattycus	0.2023	0.098	0.743
2	why did you fuck it up. ...	ElleCTF	-0.5423	0.108	0.892
3	Dude they dont finish ...	Karoli	-0.7322	0.245	0.755
4	WTF are you talking abou...	joy_wolf	-0.7739	0.341	0.659
5	I'll save you the trouble ...	mybirch	-0.8271	0.383	0.514
6	Im dead serious.Real...	coZZ	-0.7918	0.307	0.613
7	...go absolutely i...	2Hood4Hollyw...	-0.7096	0.237	0.763
8	Lmao im watching the...	mimismo	0.7947	0.108	0.608
9	LOL no he said What d...	erinx3leanne...	0.7798	0.091	0.578
10	truth on both counts that ...	pardonlauren	-0.296	0.141	0.766
11	Shakespeare nerd!	TLeC	-0.3595	0.714	0.286
12	you are SUCH a fucking do...	robrobberob...	-0.4005	0.35	0.65
13	Heh. Fuck 'em	havofwolves	-0.6958	0.742	0.258

Format Resize ☒ Background color ☒ Column min/max Save and Close Close

Index	content
11131	Hah been there. : Sa...
13062	Holy FUCK now THAT is a pa...
17693	i love youuuuuuu (:
14857	Beetles taste like a...
17105	lol how ya kno
4389	- Fuck yeah!
9827	Gone & paint that bitch's...
1737	well lady I'm all for ...
19793	why is it that another...
6640	i still feel bad. I woke ...
8162	you should come make th...
13470	you need to come visit ...
2411	hell yeah fucking lose...
17869	Where does most of your...

Format Resize ☒ Background color ☒ Column min/max Save and Close Close

X_test - Series	
Index	content
16652	what did you do last nigh...
17257	myspace for the freedom ...
12976	yeah I know. I hate holly...
16293	just stuff. i can handle...
19713	i forgot her namee s...
19916	Do you miss your past?
6578	jealous! I will still h...
16004	so do you believe that...
14032	what really happened...h...
10303	Tough love is key. Though ...
16231	Definantly not. Im a th...
8233	Ohh I feel for you I w...
17137	bravest? idk off the ...
4534	might install now that you...

Format
Resize
☒ Background color
☒ Column min/max
Save and Close
Close

y_train - Series

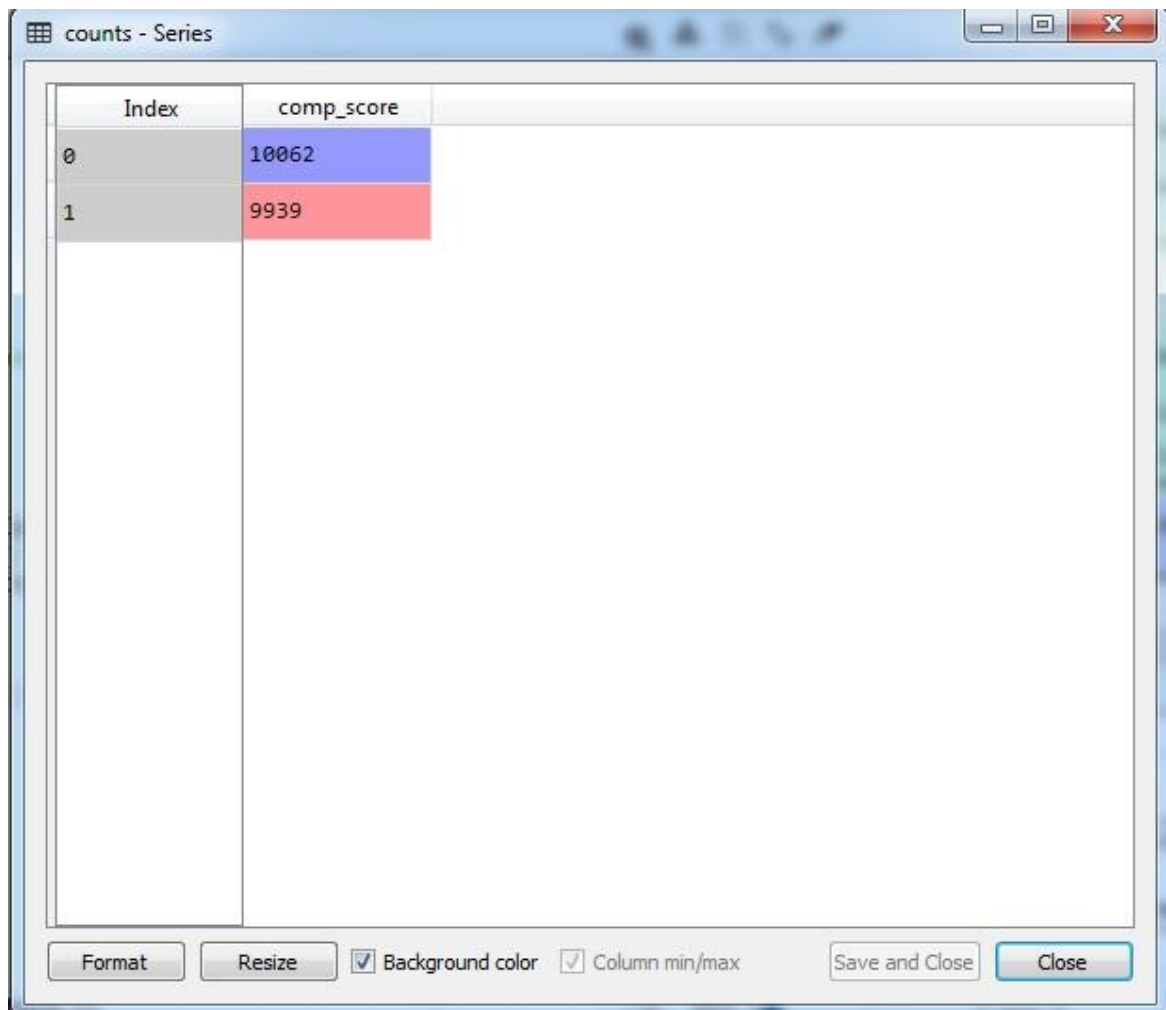
Index	comp_score
11131	1
13062	0
17693	0
14857	0
17105	0
4389	1
9827	1
1737	1
19793	1
6640	1
8162	1
13470	1
2411	1
17869	0

☒ Background color
 ☒ Column min/max

y_test - Series

Index	comp_score
16652	0
17257	0
12976	0
16293	0
19713	1
19916	1
6578	1
16004	0
14032	1
10303	1
16231	1
8233	1
17137	0
4534	0

Format Resize ☒ Background color ☒ Column min/max Save and Close Close



Index	comp_score
0	10062
1	9939

Format Resize ☒ Background color ☒ Column min/max Save and Close Close

CONCLUSIONS

9.1 Conclusions

We have developed an approach towards the detection of cyberbullying behaviour. If we are able to successfully detect such posts which are not suitable for adolescents or teenagers, we can very effectively deal with the crimes that are committed using these platforms. An approach is proposed for detecting and preventing Twitter cyberbullying using Supervised Binary classification Machine Learning algorithms. Our model is evaluated on both Support Vector Machine and Naive Bayes, also for feature extraction, we used the TFIDF vectorizer. As the results show us that the accuracy for detecting cyberbullying content has also been great for Support Vector Machine which is better than Naive Bayes. Our model will help people from the attacks of social media bullies.

9.2 Future Work

In future, it is possible to provide extensions or modifications to the proposed clustering and classification algorithms to achieve further increased performance. Apart from the experimented combination of data mining techniques, further combinations and other clustering algorithms can be used to improve the detection accuracy and to reduce the rate offensive tweets. Finally, the cyberbullying detection system can be extended as a prevention system to enhance the performance of the system.

9.3 REFERENCES

1. Amanpreet Singh, Maninder Kaur, "Content-based Cybercrime Detection: A Concise Review", International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-8, pages 1193-1207, 2019.
2. Ying Chen, Yilu Zhou, Sencun Zhu, and Heng Xu. "Detecting offensive language in social media to protect adolescent online safety". In Privacy, Security, Risk and Trust (PASSAT), 2015 International Conference on and 2012 International Conference on Social Computing (SocialCom), pages 71– 80. IEEE, 2015.
3. NaliniPriya. G and Asswini. M (2015) "A Dynamic Cognitive System For Automatic Detection And Prevention Of Cyber-Bullying Attacks", ARPN Journal of Engineering and Applied Sciences ©2006-2015 Asian Research Publishing Network (ARPN). VOL. 10, NO. 10, JUNE 2015.
4. "Protective shield shield for social networks to defend cyberbullying and online grooming attacks" in Proceedings of 40th IRF International Conference, Pune, India, ISBN: 978-93-85832-16-1, 2015.
5. Gupta, P. Kumaraguru, and A. Sureka, "Characterizing pedophile conversations on the internet using online grooming," CoRR, vol. abs/1208.4324, 2012.
6. K. Jedrzejewski and M. Morzy, "Opinion Mining and Social Networks: A Promising Match," 2011 Int. Conf. Adv. Soc. Networks Anal. Min., pp. 599– 604, Jul. 2011.
7. H. Hosseinmardi, S. A. Mattson, R. I. Rafiq, R. Han, Q. Lv, and S. Mishra, "Analyzing Labeled Cyberbullying Incidents on the Instagram Social Network."
8. Kelly Reynolds, April Kontostathis, Lynne Edwards, "Using Machine Learning to Detect Cyberbullying", 2011 10th International Conference on

Paper Publication

1. WJARR

World journal of advanced research and reviews.

Issue Date	Paper Title	Author Name	Description
28, April 2022	Cyberbullying Detection in Social Media Using Supervised ML and NLP techniques. Published at World journal of advanced research and reviews (WJARR).	Gayatri Battise, Anuj Patil, Anklesh Patil, Jivan Devhare	We have developed an approach towards the detection of cyberbullying behaviour. If we are able to successfully detect such posts which are not suitable for adolescents or teenagers, we can very effectively deal with the crimes that are committed using these platforms. An approach is proposed for detecting and preventing Twitter cyberbullying using Supervised Binary classification Machine Learning algorithms. Our model is evaluated on both Support Vector Machine and Naive Bayes, also for feature extraction, we used the TFIDF vectorizer. As the results show us that the accuracy for detecting cyberbullying content has also been great for Support Vector Machine which is better than Naive Bayes. Our model will help people from the attacks of social media bullies.



WJARR

World Journal of Advanced Research and Reviews

An International, Peer reviewed, Open access Journal

Published by GSC Online Press

eISSN:2581-9615 CODEN : WJARAI



This is to certify that

Mr. Gayatri Battise has contributed as an author of

Entitled

Cyberbullying Detection in Social Media Using Supervised

published in World Journal of Advanced Research and Reviews (WJARR)

Volume 1-II Issue 12 of year April 2022.

Certificate No: WJARR399425

G. S. Chaudhary

Editor in chief

DOI: 10.5612/2356.2022.01.0134



WJARR

World Journal of Advanced Research and Reviews

An International, Peer reviewed, Open access Journal

Published by GSC Online Press

eISSN:2581-9615 CODEN : WJARAI

This is to certify that

Mr. Anuj Patil has contributed as an author of

Entitled

Cyberbullying Detection in Social Media Using Supervised

published in World Journal of Advanced Research and Reviews (WJARR)

Volume 1-II Issue 12 of year April 2022.

Certificate No: WJARR399425

G. S. Chaudhari

Editor in chief



DOI: 10.5612/2356.2022.01.0134



WJARR

World Journal of Advanced Research and Reviews

An International, Peer reviewed, Open access Journal

Published by GSC Online Press

eISSN:2581-9615 CODEN : WJARAI



This is to certify that

Mr. Jivan Devhare has contributed as an author of

Entitled

Cyberbullying Detection in Social Media Using Supervised

published in World Journal of Advanced Research and Reviews (WJARR)

Volume 1-II Issue 12 of year April 2022.

Certificate No: WJARR399425

G. S. Chaudhari

Editor in chief

DOI: 10.5612/2356.2022.01.0134



WJARR

World Journal of Advanced Research and Reviews

An International, Peer reviewed, Open access Journal

Published by GSC Online Press

eISSN:2581-9615 CODEN : WJARAI

This is to certify that

Mr. Anklesh Patil has contributed as an author of

Entitled

Cyberbullying Detection in Social Media Using Supervised

published in World Journal of Advanced Research and Reviews (WJARR)

Volume 1-II Issue 12 of year April 2022.

Certificate No: WJARR399425

G. S. Chaudhari

Editor in chief



DOI: 10.5612/2356.2022.01.0134