SRS(Software Requirement Specification)

# Project Title:

**Cyberbullying detection on social media using machine learning Submitted by:**

Harsh Agarwal Jagruti Jadhav Babita Jaybhaye Komal Nagar

## UNDER THE GUIDANCE OF

Prof. Shrikant Dhamdhere

B.E.(COMPUTER SCIENCE)

1. Introduction

1.1 Purpose:

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

1.2 Scope:

The study will show the effects and the consequence of cyberbullying today in our society, especially to those teenagers who are most affected. This will answer the question about how we can prevent

from being cyberbullied. This study will show us how teens are cyberbullied and the fact that being a victim of cyberbullying can be a common and painful experience.

Early detection of harmful social media behaviors such as cyberbullying is necessary for identifying threatening online abnormalities and preventing them from increasing. So, In this project we successfully fetched the comments from the subreddit using praw, and I was also able to identify the vulgar comments by using three machine learning algorithms: bag of words, term frequency inverse document frequency, support vector machine.

2. Overall Description

2.1 Proposed system:

Cyberbullying detection is designed using machine learning techniques. Twitter data set is collected with features and labels and mode is trained using **the SVM** algorithm and trained model is applied to live chatting application which has multiple clients and a single server. For each message, cyberbullying is detecting using the model and then alert messages are posted on chat boar

2.2 Product features:

* Cyberbullying detection process is automatic and time taken for detection is less and it works on the live environment.
* The latest machine learning models are used for training models that are accurate.

2.3 User classes and characteristics:

2.4 Operating Environment

**Software tools:**

GitHub Desktop, MlFlow, DVC, Tox, Postman, Heroku

Hardware specification:

Windows 8, 8Gb ram, Intel I5 processor.

2.5 Design and Implementation constraints

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. System Features

3.1 Functional Requirements of the System:

Following are the functional requirements on the system:

**Comment prediction requirement**

* The system should provide text ­to ­feature function which can take the necessary part and obtain a feature vector**.**
* The system should have a well-trained SVM to generate better inputs for classifier.
* The system should provide text parser functions which can take the whole text and separate into tokens.
* The system needs a classifier which is well­ trained that predicts the probability of each sentence.

**Web page requirement**

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

**Train System Requirements**

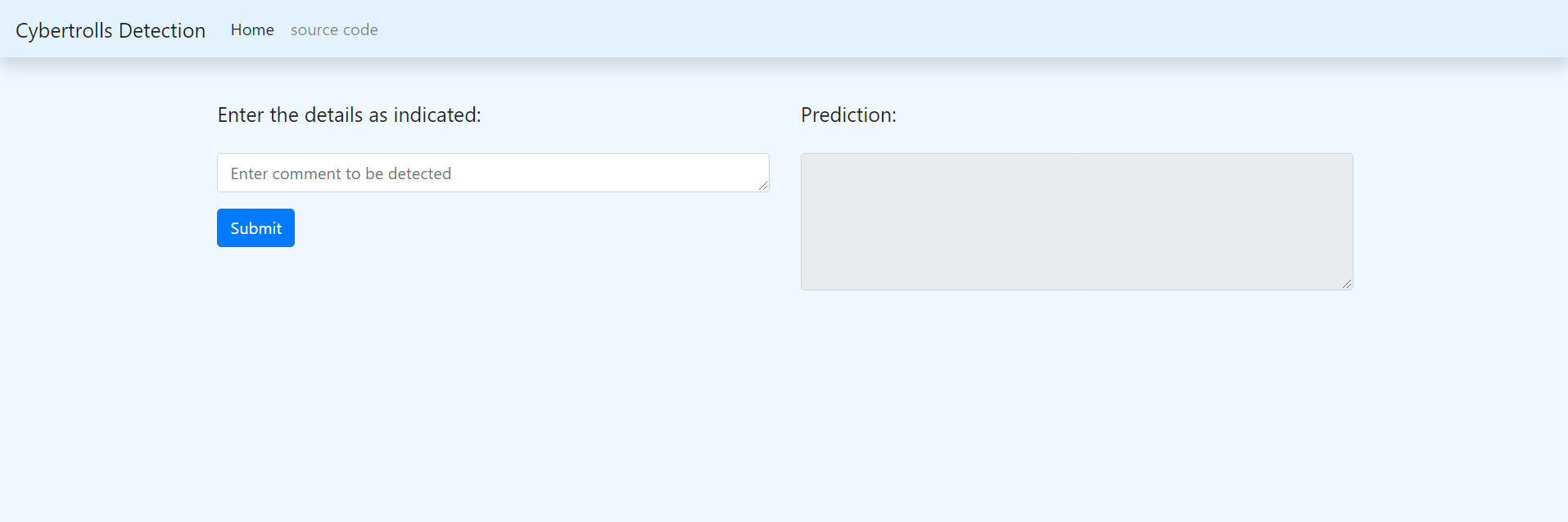
* The system should provide a configuration file for taking new data from admin to train models

4. External Interface requirement

4.1 User interfaces:

User interface had a submit button. When user clicks submit button on a webpage it

triggers the prediction function and in the text box it gives the prediction of sentence

The prototype user interface is as follows:

4.2 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.

4.3 Communication interface

The only communication 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

4.4 Hardware interface:

Not applicable

5. Non-Functional Requirements of the System:

#### Correctness:

1. Model should have accuracy between 60-90%
2. Data privacy should be maintained

#### Ease of Use:

1. Algorithm should not take more than 10 mins for execution
2. Model Coder provides an interface which allows the user to interact in an easy manner

* **Modularity:** The complete product is broken up into many modules and well-defined interfaces are

developed to explore the benefit of flexibility of the product

* **Robustness:** This software is being developed in such a way that the overall performance is optimized and the user can expect the results within a limited time with utmost relevancy and correctness. Non functional requirements are also called the qualities of a system. These qualities can be divided into execution quality & evolution quality. Execution qualities are security & usability of the system which are observed during run time, whereas evolution quality involves testability, maintainability, extensibility or scalability