CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Cyberbullying is a supervised learning problem, where the system is trained with data labelled by humans as cyberbullying. Later, the trained system can be used to detect cyberbullying automatically. The first step in cyberbullying detection is to learn a reliable numeric representation for the text messages. This is achieved through a word2vec model of insulting seeds and use of cosine similarity to extend the seeds to a larger set. The insulting seeds are the set of words which can be called as the bullying words and its extensions (different spellings, repetitions etc.). This constructed bullying features are then given to a stacked denoising auto encoder. The output of the autoencoder provides robust and discriminative features, which is used with a support vector machine (SVM) classifier. The classifier performs a binary classification and detects if the given message is bullying or not. The dataset used in this project is openly available and is taken from social networking websites such as Myspace or Twitter.

1.2 PROBLEM DEFINITION

Social networking and online chatting applications provide a platfom for any user to share knowledge and talent but few users take this platform to threaten users with cyberbullying attacks which causes issues in using these platforms.

1.3 PROJECT PURPOSE

The scope of the project is that, it gives a specialized representation learning model for cyber bullying detection and also gives a robust and discriminative representation for detection .

1.4 SCOPE OF THE PROJECT

Cyberbullying is a major problem and has been documented as a serious national health problem due to the recent growth of online communication and social media websites. Research has shown that cyberbullying exerts negative effects on the psychological and physical health and academic performance of people. Cyberbullying victims incur a high risk of suicidal ideation.

Consequently, developing a cyberbullying prediction model that detects aggressive behaviour that is related to the security of human beings is more important than developing a prediction model for behaviour related to the security of machines. Social media websites have become an integral part of user’s lives a study found that social media websites are the most common platforms for cyberbullying victimization detection of such cyberbullying attacks becomes absolutely necessary.

1.5 PROJECT FEATURES

Myspace is a social networking website which is similar to twitter. There will be a series of messages that are captured and stored as a dataset. The dataset will be read, cleaned (to remove noises like special characters, stop words (words like this, to, was)), preprocessed (create numerical representation for each word) using a model that we create (word2vec).
Deep learning algorithms are neural networks that are helpful in learning complex underlying representations. A neural network will be trained on the preprocessed data which is trained to learn the features of bullying data and non-bullying data. Classification will be performed using a classifier algorithm (such as Support Vector Machine). Support Vector Machine is a linear classifier capable of separating two classes of data (bullying/non-bullying) based on the given features.
In Python Open CV library is available for image processing. The Open CV is a library of programming functions mainly aimed at real-time computer vision.

1.6 MODULE DESCRIPTION

In this project there are six modules to achieve our expected result. These are the major functionalities of the project.

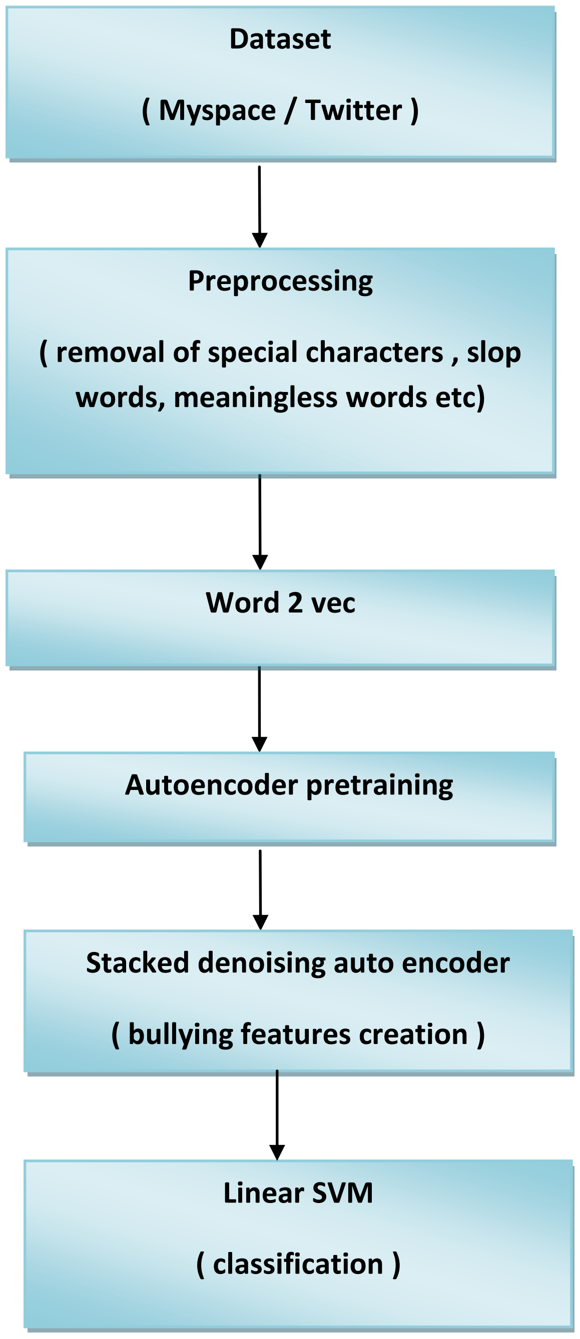

Dataset
(Myspace / Twitter)
Preprocessing
(removal of special characters , slop words, meaningless words etc)
Word 2 vec
Autoencoder pretraining
Stacked denoising auto encoder
(bullying features creation)
Linear SVM
(classification)

Fig 1.1 Architecture Diagram

CHAPTER 2

LITERATURE SURVEY

Literature survey is the most important step in software development process. Before developing the tool, it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, ten next steps are to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system, the above consideration is taken into account for developing the proposed system.

2.1 RELEATED WORK

This work aims to learn a robust and discriminative text representation for cyberbullying detection. Text representation and automatic cyberbullying detection are both related to our work. In the following, we briefly review the previous work in these two areas.

2.2 TEXT PRPRESENTATION LEARNING

In text mining, information retrieval and natural language processing, effective numerical representation of linguistic units is a key issue. The Bag-of-words (BoW) model is the most classical text representation and the cornerstone of some states-of- arts models including Latent Semantic Analysis (LSA) and topic models. BoW model represents a document in a textual corpus using a vector of real numbers indicating the occurrence of words in the document. Although BoW model has proven to be efficient and effective, the representation is often very sparse. To address this problem, LSA applies Singular Value Decomposition (SVD) on the word-document matrix for BoW model to derive a low-rank approximation. Each new feature is a linear combination of all original features to alleviate the sparsity problem. Topic models, including Probabilistic Latent Semantic Analysis and Latent Dirichlet Allocation, are also proposed. The basic idea behind topic models is that word choice in a document will be influenced by the topic of the document probabilistically.
Topic models try to define the generation process of each word occurred in a document. Similar to the approaches aforementioned, our proposed approach takes the BoW representation as the input. However, our approach has some distinct merits. Firstly, the multi-layers and non-linearity of our model can ensure a deep learning architecture for text representation, which has been proven to be effective for learning high-level features. Second, the applied dropout noise can make the learned representation more robust. Third, specific to cyberbullying detection, our method employs the semantic information, including bullying words and sparsity constraint imposed on mapping .

2.3 DATA ABSTRACTION

For the included papers, we performed data abstraction using characteristics such as detection tasks performed, data sources, the size and availability of the datasets, detection techniques, annotation judgement, features extracted, external resources used, and pre-processing steps. We used the total number of documents (i.e., messages, posts, comments, etc.) as a measure of the data size as opposed to using other metrics such as the number of users or threads in a dataset; thus, a sample containing 50 messages generated by 70 users was assigned 50 as the data size value. etc.). Table 3 presents details (where available) of the datasets used by the papers. Finally, the best available results per detection tasks for each corpus category are presented in Ta

2.4 EXITING SYSTEM

There are many cyber bullying algorithms available in the market most of which are supervised algorithms. These types of algorithms generally use bag of words to distinguish between bullying and non-bullying labels.

DISADVANTAGES

They cannot corelate similar words. For example, if an algorithm uses bag of words it will classify cat and kitten as two different entities instead of classifying them as related entities unless explicitly implied.

2.2.1 GRAPHICAL USER INTERFACE (GUI)

The Tk interpackage ("Tk interface") is the standard Python interface to the Tk GUI toolkit. Both Tk and Tk inter are available on most Unix platforms, as well as on Windows systems. (Tk itself is not part of Python; it is maintained at ActiveState.)
Running python Tk inter from the command line should open a window demonstrating a simple Tk interface.

Here are some recent research papers on cyberbullying

Detecting cyberbullying on social media: A machine learning approach" by N. D. Dinh et al. (2021) This study proposed a new approach to cyberbullying detection on social media using machine learning algorithms, and compared the performance of different classifiers on a large dataset of Twitter messages.

"A hybrid approach to cyberbullying detection using machine learning and deep learning techniques" by A. Alqadomi and M. Aljarah (2020). This paper proposed a hybrid approach combining machine learning and deep learning techniques for cyberbullying detection on social media, and compared the performance of different models on a dataset of Arabic tweets.

"A deep learning approach to cyberbullying detection on Instagram" by S. S. Sood et al. (2019). This study proposed a deep learning approach for cyberbullying detection on Instagram, and compared the performance of different models on a dataset of user comments.

Overall, these papers highlight the importance of developing effective techniques for cyberbullying detection on social media, and demonstrate the potential of machine learning and deep learning methods for this task. However, further research is needed to improve the accuracy and scalability of these methods, and to address ethical and privacy concerns related to cyberbullying detection.

CHAPTER 3

REQUIRMENT ANALYSIS

3.1 FUNCTIONAL REQUIREMENTS

The functional requirements in cyber bullying are:

3.1.1 Commenting on Posts

3.1.2 Reporting and Blocking

Commenting on Posts: Users can make comments on the posts of other users who they are following. Reporting : Users can report about other users if they find them annoying or disturbing.

3.2 NON-FUNCTIONAL REQUIREMENTS

Nonfunctional requirements describe how a system must behave and establish constraints of its functionality. This type of requirements is also known as the system's quality attributes. Attributes such as performance, security, usability, compatibility are not the feature of the system, they are a required characteristic. They are "developing" properties that emerge from the whole arrangement and hence we can't compose a particular line of code to execute them.
Any attributes required by the customer are described by the specification. We must include only those requirements that are appropriate for our project.

Some Non-Functional Requirements are as follows:

- Reliability
- Maintainability
- Performance
- Portability
- Scalability
- Flexibility

3.1.1 ACCESSIBILITY

Accessibility is a general term used to describe the degree to which a product, device, service, or environment is accessible by as many people as possible.

3.1.2 MAINTAINABILITY

In software engineering, maintainability is the ease with which a software product can be modified in order to:

- Correct defects
- Meet new requirements

New functionalities can be added in the project based on the user requirements.

.

3.1.3 SCALABILITY

System is capable of handling increase total throughput under an increased load when resources (typically hardware) are added System can work normally under situations such as low bandwidth and large number of users.

3.1.4 PORTABILITY

Portability is one of the key concepts of high-level programming. Portability is the software code base feature to be able to reuse the existing code instead of creating new code when moving software from an environment to another.

Project can be executed under different operation conditions provided it meets its minimum configurations. Only system files and dependant assemblies would have to be configured in such case.

3.1.5 RELAIBILITY

Software Reliability is the probability of failure-free software operation for a specified period of time in a specified environment. Software Reliabilitysis also an important factor affecting system reliability.

3.3 HARDWARE REQUIREMENTS

System	**:** Pentium IV 2.4 GHz.
Hard Disk	**:** 40 GB.
Floppy Drive	**:** 1.44 Mb.
Monitor	: 14' Colour Monitor.
Mouse	**:** Optical Mouse.
Ram	**:** 4GB.

3.4 SOFTWARE REQUIREMENTS

Operating system	: Windows XP,7,8,8.1,10
Coding language	: Python
Front end	: Python
IDE	: Jupyter
Libraries	: pandas , streamlit , nltk , matplotlib

3.5 SYSTEM REQUIREMENT SPECIFICATION DOCUMENT

Software Requirement Specification (SRS) is the starting point of the software developing activity. As system grew more complex it became evident that the goal of the entire system cannot be easily comprehended. Hence the need for the requirement phase arose. The software project is initiated by the client needs. The SRS is the means of translating the ideas of the minds of clients (the input) into a formal document (the output of the requirement phase.

THE SRS PHASE CONSIST OF TWO BASICS ACTIVITIES

1. PROBLEM / REQUIREMENTS ANALYSIS

The process is order and more nebulous of the two, deals with understanding the problem , the goal and constraints.

2. REQUIREMENTS SPECIFICATION

Here, the focus is on specifying what has been found giving analysis such as representation, specification languages and tools, and checking the specifications are addressed during this activity. The Requirement phase terminates with the production of the validate SRS document. Producing the SRS document is the basic goal of this phase.

ROLE OF SRS

The purpose of the Software Requirement Specification is to reduce the communication gap between the clients and the developers. Software Requirement Specification is the medium through which the client and user needs are accurately specified. It forms the basis of software development. A good SRS should satisfy all the parties involved in the system.

CHAPTER 4

SYSTEM ANALYSIS

4.1 PROPOSED SYSTEM

The proposed system will make cyber bullying detection more dynamic i.e. Instead of using bag of words algorithm it will use word2vec model which is a multi-dimensional algorithm that can plot and distinguish between related entities without being explicitly programmed. This will play a great role in enhancing the dynamic of cyber bullying detection algorithm. Hence this will improve the performance of any of the existing models.

ADVANTAGES

- Data mining techniques are to analysis the student performance.
- Portability

4.2 SYSTEM STUDYS

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are ,

- **ECONOMICAL FEASIBILITY**
- **TECHNICAL FEASIBILITY**
- **SOCIAL FEASIBILITY**

• ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

Economical feasibility using COCOMO model :

	AB	BB	CB	DB
O	4.4	2.02	1.5	0.26
SEMIDETACHED	2.0	3.12	1.5	0.23
EMBEDDED	3.4	2.10	1.5	0.34

CALCULATION :

KLOC= 3.0

Effort per Month (E) = ab*(KLOC)^bb

= 3.4 *(3.0)^ 2.10

E = 34.15(approx.)

Person – Month

Development Time(D) =cb*(E)^db

=1.5 *(13.15)^ 0.34

D =3.60 month

No. of Team Members = E/D

= 13.15 / 3.60

= 3.65

= ~ 3 person's approx.

For personal cost we did the following calculations:

Generally the computer engineer has Rs.15000 salary per month, therefore

1 month = Rs.15000

Per day he works only 8 hours

Therefore 30*8=240 hours

He works 240 hours per month

Therefore,

15000/240=Rs62.5 for 1 hour

We are three project partners. We work for 5 weeks a month

• TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

HARDWARE COSTS

SYSTEM	Pentium IV 2.4 GHs	5000/-
HARD DISK	40GB	1800/-
FLOPPY DISK	1.44Mb	1500/-
MONITOR	14’Color Monitor	10,000/-
MOUSE	Optimal Mouse	500/-
RAM	4 GB	4000/-

Table 4.1 Hardware Costs

SOFTWARE COSTS

PYTHON	5000/-
NUMPY PANDAS	**FREE**

Table 4.2 Software Costs

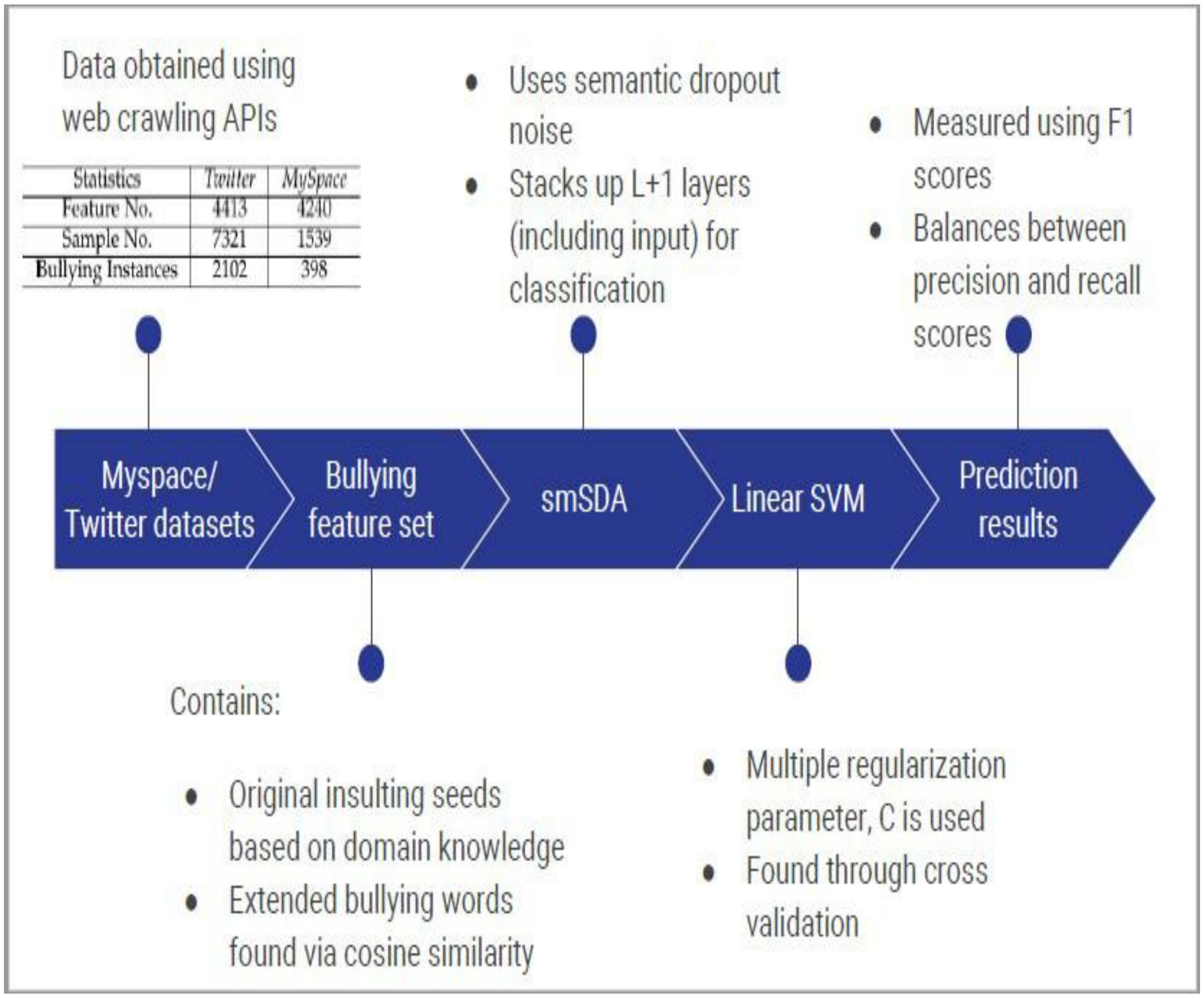
• SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER 5

DESIGN

5.1 WORKFLOW


Data obtained using web crawling APIs
Myspace/ Twitter datasets
Bullying feature set
smSDA
Linear SVM
Prediction results
• Uses semantic dropout noise
• Stacks up L+1 layers (including input) for classification
• Measured using F1 scores
• Balances between precision and recall scores
Contains:
• Original insulting seeds based on domain knowledge
• Extended bullying words found via cosine similarity
• Multiple regularization parameter, C is used
• Found through cross validation

Statistics	Twitter	MySpace
Feature No.	4413	4240
Sample No.	7321	1539
Bullying Instances	2102	398

Fig 5.1 Workflow

5.2 ARCHITECTURE DIAGRAM

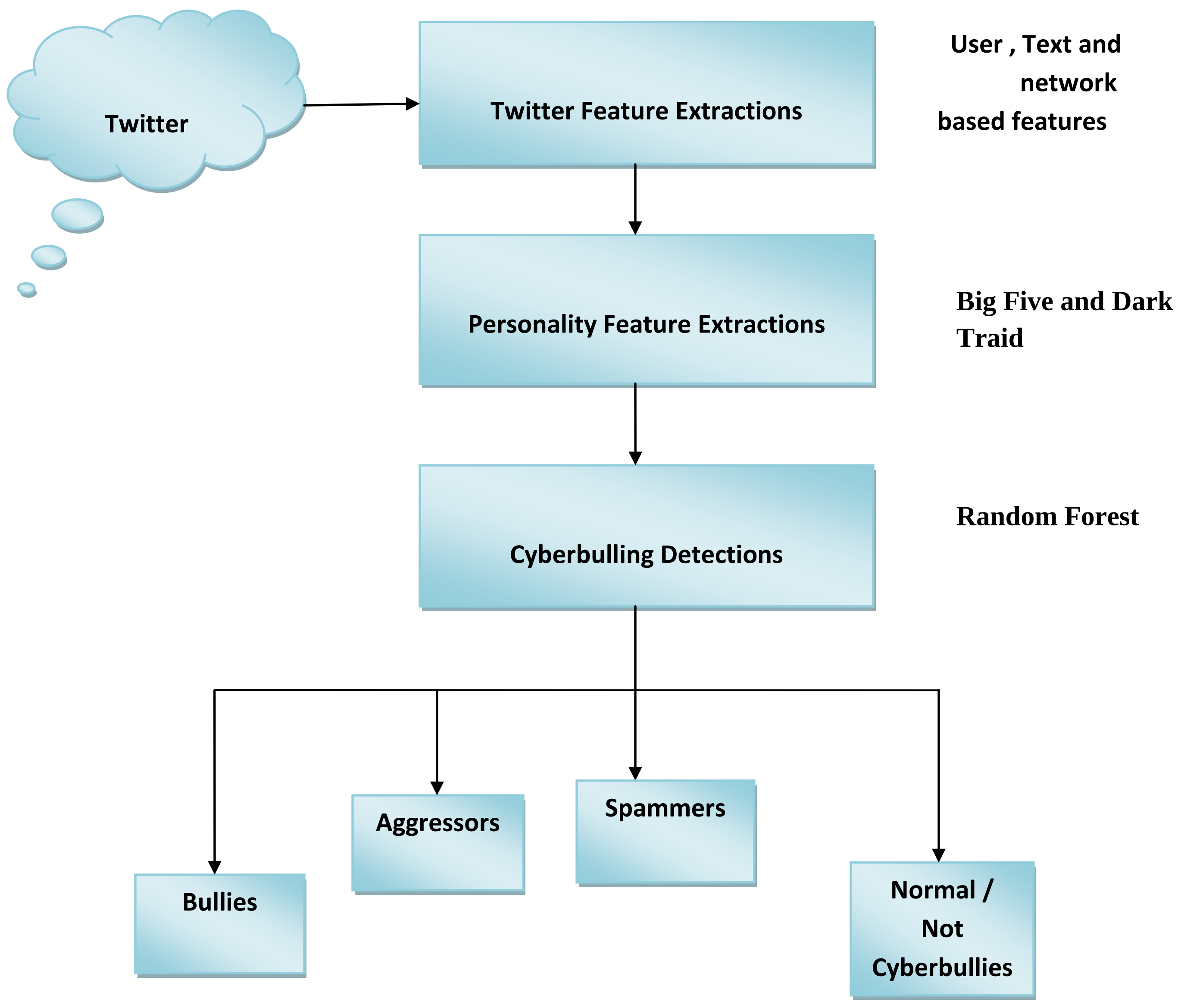

Twitter
Twitter Feature Extractions
User , Text and network based features
Personality Feature Extractions
Big Five and Dark Traid
Cyberbulling Detections
Random Forest
Bullies
Aggressors
Spammers
Normal / Not Cyberbullies

Fig 5.2 Architecture Diagram

5.3 USER MODULE

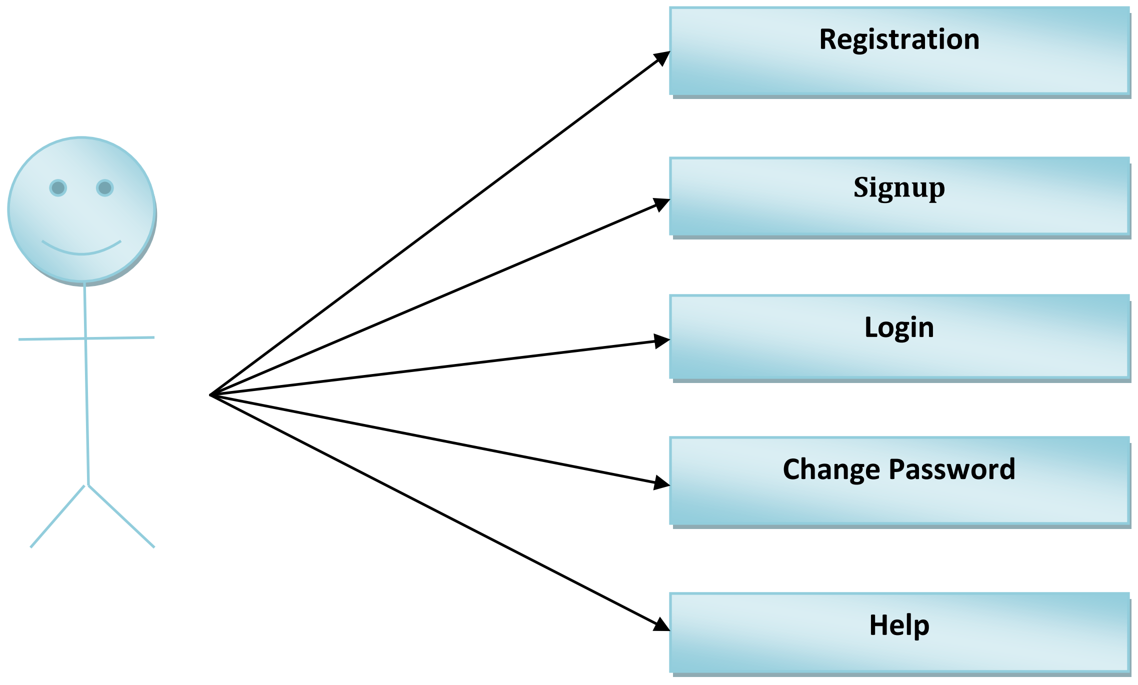


Fig 5.3 User Case Diagram

The user has lots of features and functionality to do. User can sign up to the web application by registrering themselves by providing details like user name, password etc.. Registered users can also sign in to their profile by using id and password.

If user have any complaint or need, we also add feature of help section where user send their complaints on web application.

The application is reduced as much possible to avoid errors while entering the data. It also provides error message while entering invalid data. No formal knowledge is needed for the user to use this system. This, this all it user-friendly.

5.4 MACHINE LEARNING MODULE

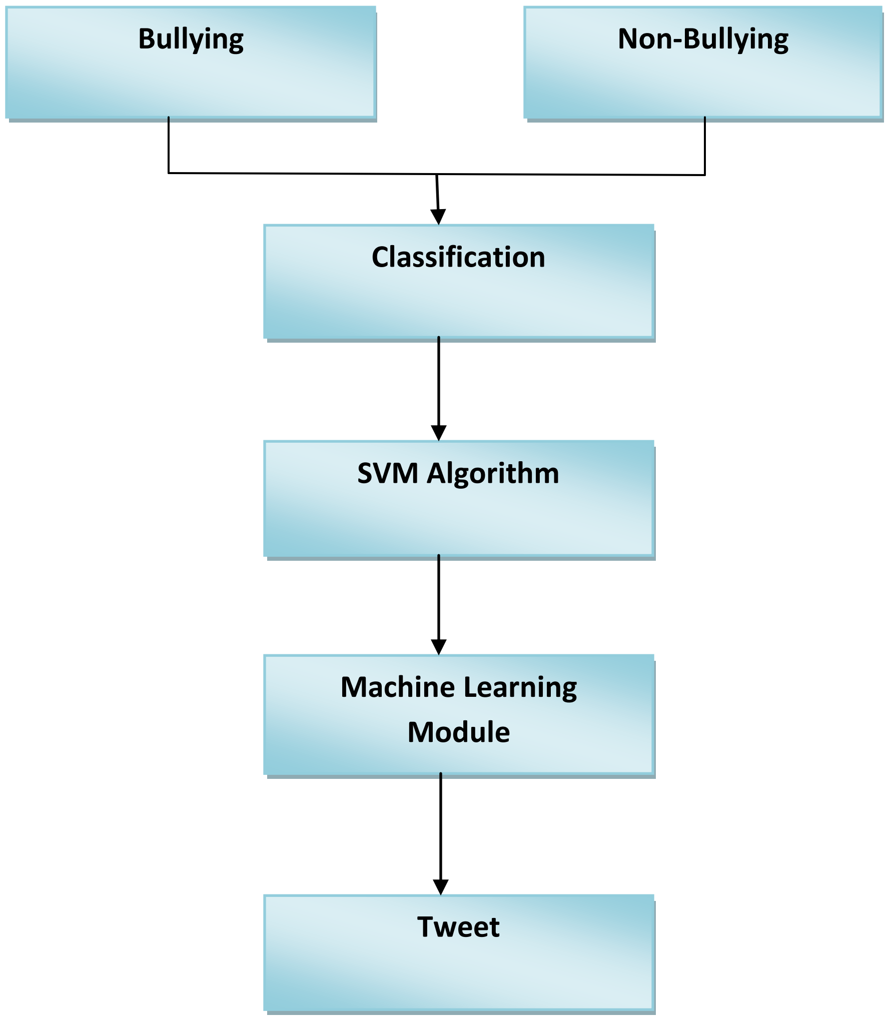


Fig 5.4 Machine Learning Use Case Diagram

The Machine Learning module is responsible for classifying comments and messages as bullying or non-bullying. From huge set of comments and messages.

This module includes the following steps:

- Data collection
- Data preparation
- Segmentation
- Feature extraction
- Training
- Testing

Data collection: Collecting data for training the Machine Learning module is the basic step in the machine learning.

Data preprocessing: Data preprocessing is Machine Learning is crucial step helps the quality of data to promote the extraction of meaningful insights from the data.

Segmentation: Segmentation can be defines as the process of separating sentences into different tokens.

Feature extraction: Feature extraction is the process of taking out a list of words from the text data and then transforming them into a feature set which is usable by a classifier.

Training: Model training is the key step in machine learning that results in a model ready to be validated, tested, and deployed.

Testing: In machine learning, testing model is referred to as the process where the performance of a fully trained model is evaluated on a testing set.

5.5 E-R DIAGRAM

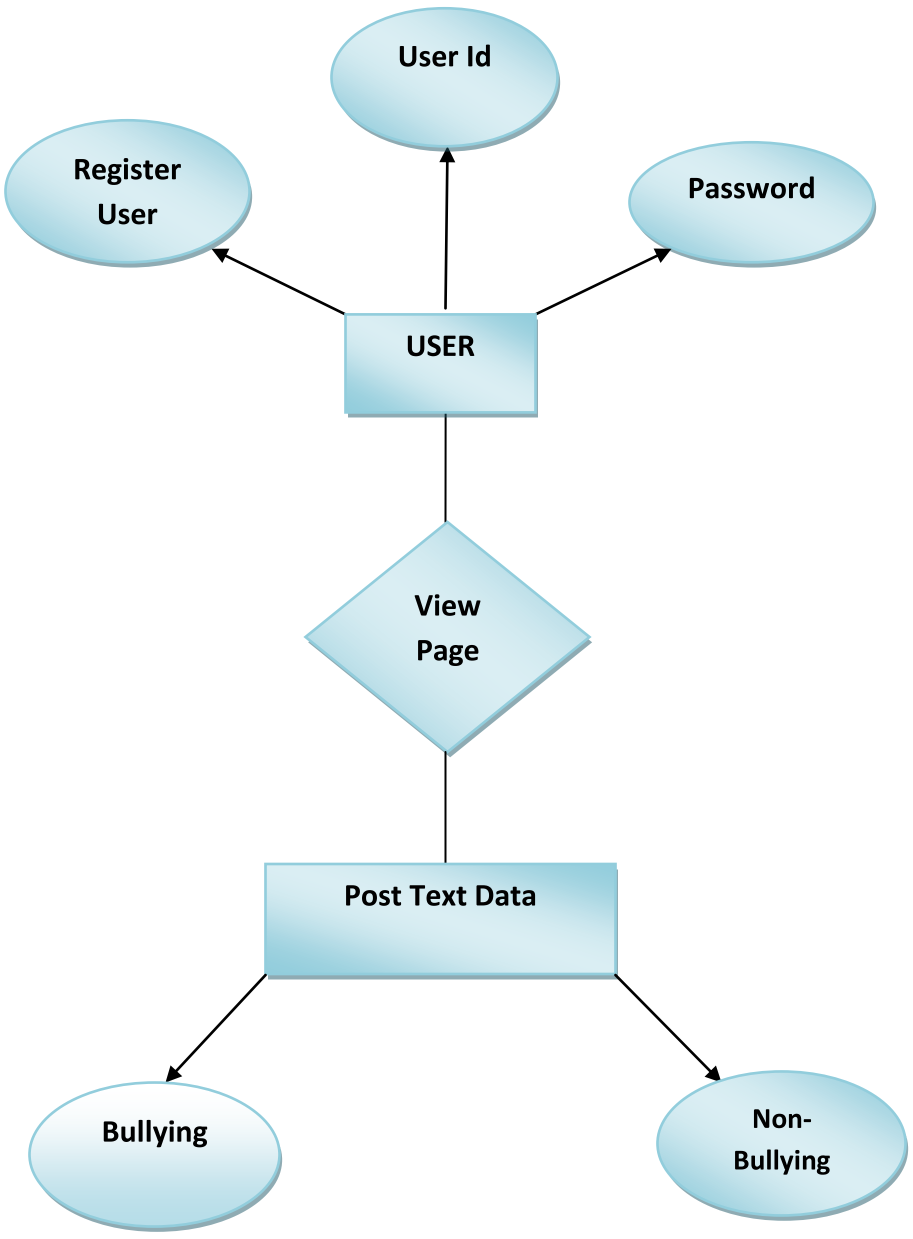


Fig 5.5 ER Diagram

An entity relation (ER) diagram is specialized graphics that illustrates the inter relationship between entities in a database. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities.

Diamonds are normally used to represent relationships and ovals are used to represent attributes. An entity relationship model (ERM), in software engineering is an abstract and conceptual representation of data. ERM is a relational schema database modelling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion.

1. Entity: Entity is the thing which we want to store information. It is an elementary basic building block of storing information about business processes. An entity represents an object defined within the information system about which you want to store information. Entities are distinct things in the enterprise.

2. Relationships: A relationship is a named collection or association between entities used to relate two or more entities with some common attributes or meaningful interaction between the objects.

3. Attributes: Attributes are the properties of the entities and relationship, descriptors of the entity. Attributes are elementary pieces of information attached to an entity.

5.6 ACTIVITY DIAGRAM

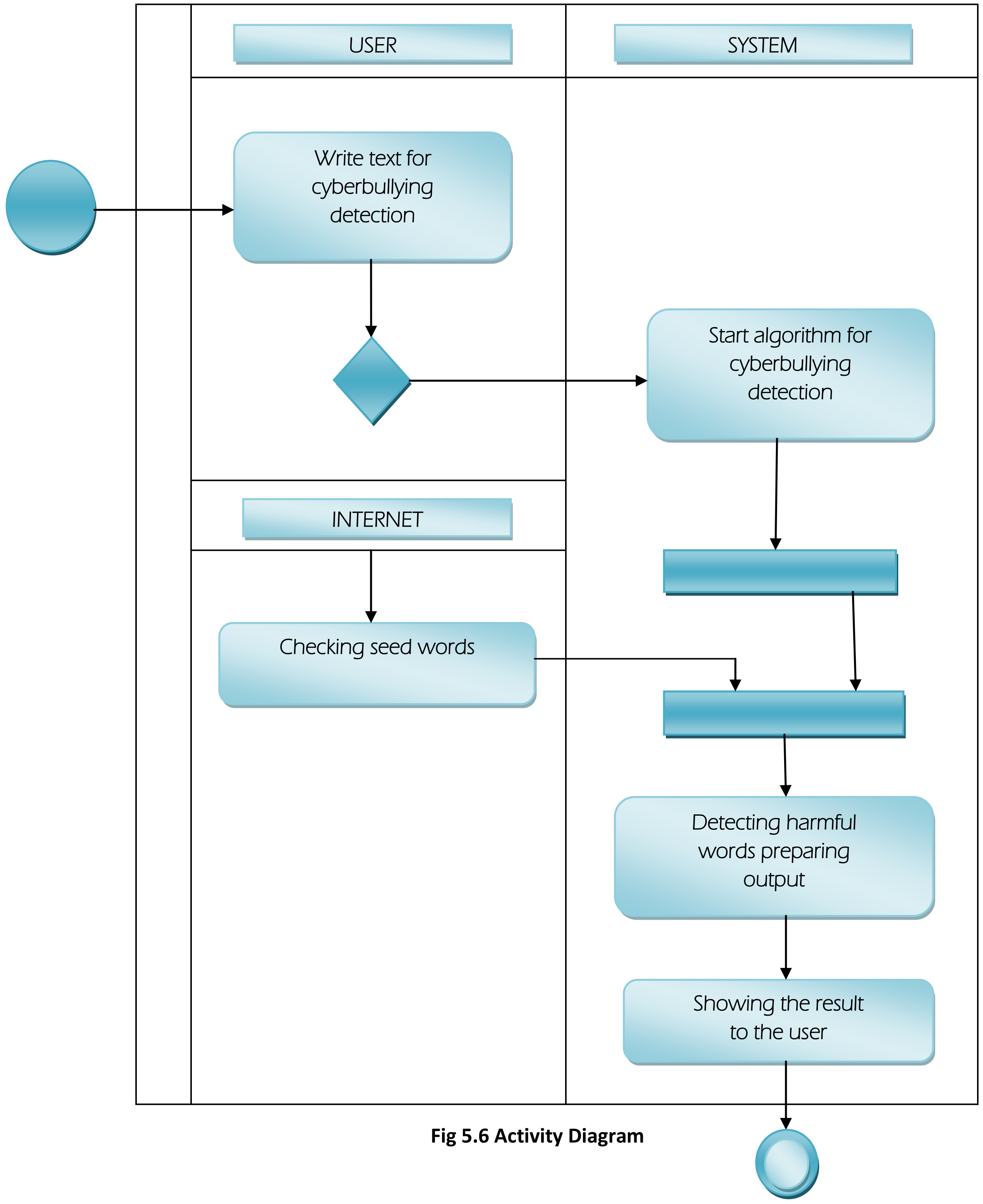

USER
SYSTEM
Write text for cyberbullying detection
Start algorithm for cyberbullying detection
INTERNET
Checking seed words
Detecting harmful words preparing output
Showing the result to the user

Fig 5.6 Activity Diagram

CHAPTER 6

IMPLEMENTATION

PREPROCESSING

- **Extract Labels :** Here we club the posts related to a specific topic and group them in series of ten or less than ten and store them in a file.
- **Extraction of information from xlsx files and formation of data frame :** Each packet contains around 200 files and we have 10 packets. Each file is tested for cyberbullying and marked accordingly.
- **Function for clean-up of the Myspace data :** Pre-Processing for the data by removing stop words, repeating words and special characters is done here.

6.1 PYTHON

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++or Java.

It provides constructs that enable clear programming on both small and large scales. Python interpreters are available for many operating systems. Python, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations.

Python is managed by the non-profit Python Software Foundation. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

6.2 SVM

SVM (support vector machine) is a supervised machine learning algorithm. That's why training data is available to train the model. SVM uses a classification algorithm to classify a two group problem. SVM focus on decision boundry and support vectors, which we will discuss in the next section. SVM is very powerful compared to other algorithms.

6.3 JUPYTER NOTEBOOK

The jupyter notebook is a server client application that allows editing and running notebook documents via a web browser. The jupyter notebook can be executed on a local desktop requiring no internet access or can be installed on a remote server and accessed through the internet.

In addition to displaying/editing/running notebook documents , the jupyter notebook app has a "Dashboard" (Notebook Dashboard) , a "control panel" showing local files and allowing to open notebook documents or shutting down their kernels.

6.4 PYTHON JSON

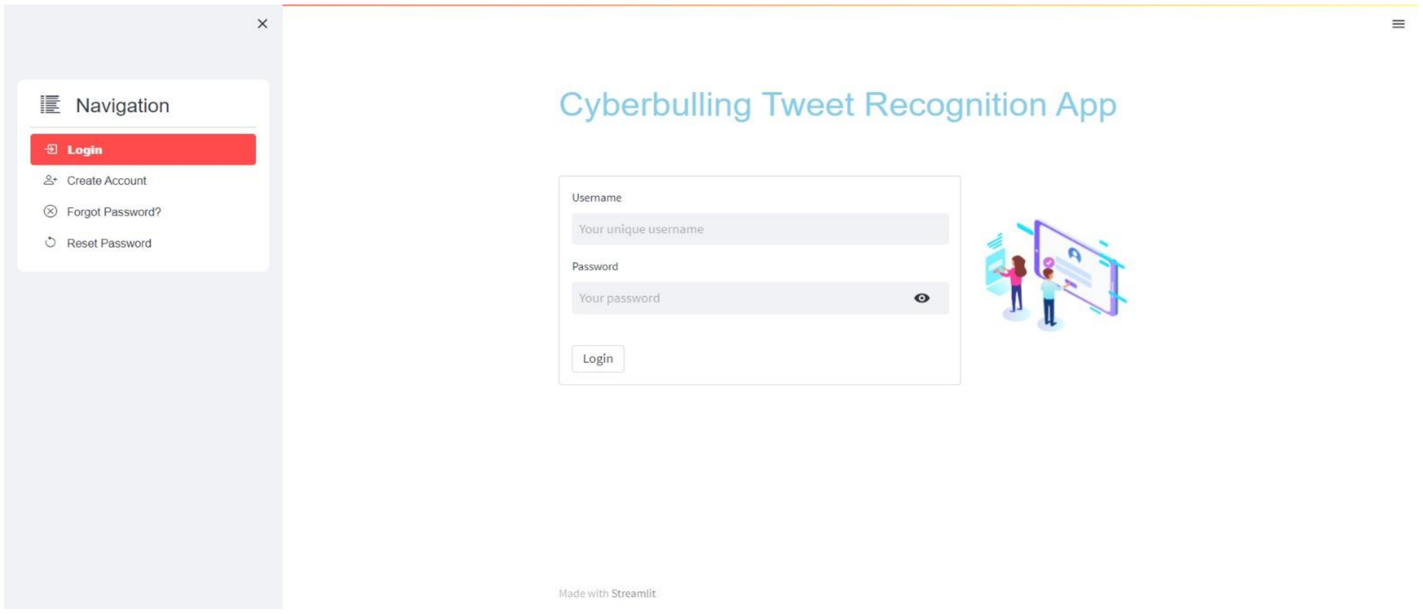
Python JSON Java script Object Notation is a format for structuring data. It is mainly used for storing and transferring data between the browser and the server. Python too supports JSON with a built-in-package called JSON. This package provides all the necessary tools for working with JSON objects including parsing, serializing, deserializing, and many more.

6.5 KAGGLE

Kaggle is an online community platform for data scientists and machine learning enthusiasts. Kaggle allows users to collaborate with other users, find and publish datasets, use GPU integrated notebooks, and compete with other data scientiststo solve data science challenges, The aim of this online platform is to help professionals and learners reach their goals in their data science journey with the powerful tools and resources it provides.

SCREENSHOT

- **Login Page**



Cyberbulling Tweet Recognition App

Username

Your unique username

Password

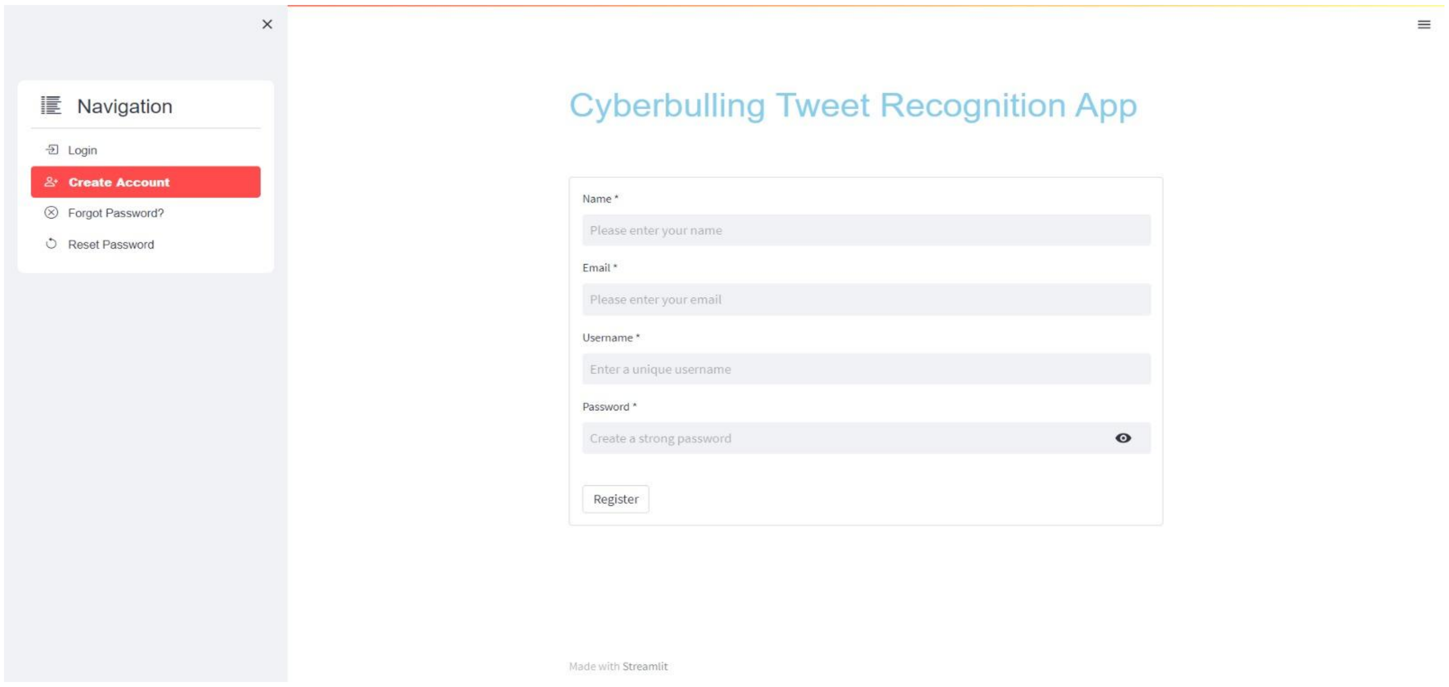
Your password

Login

Made with Streamlit

Fig 6.1 Login Page

- **Signup Page**



Cyberbulling Tweet Recognition App

Name *

Please enter your name

Email *

Please enter your email

Username *

Enter a unique username

Password *

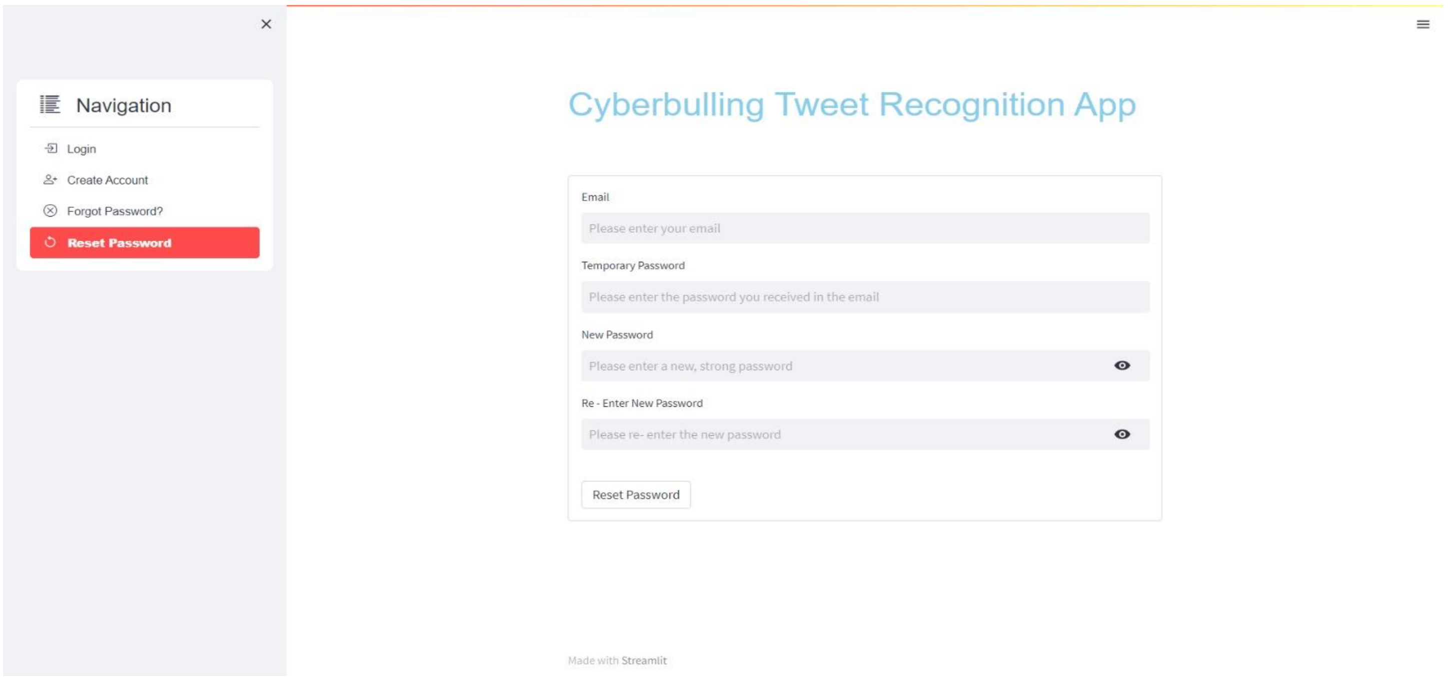
Create a strong password

Register

Made with Streamlit

Fig 6.2 Signup Page

- **Reset Password Page**



Cyberbulling Tweet Recognition App

Email

Please enter your email

Temporary Password

Please enter the password you received in the email

New Password

Please enter a new, strong password

Re - Enter New Password

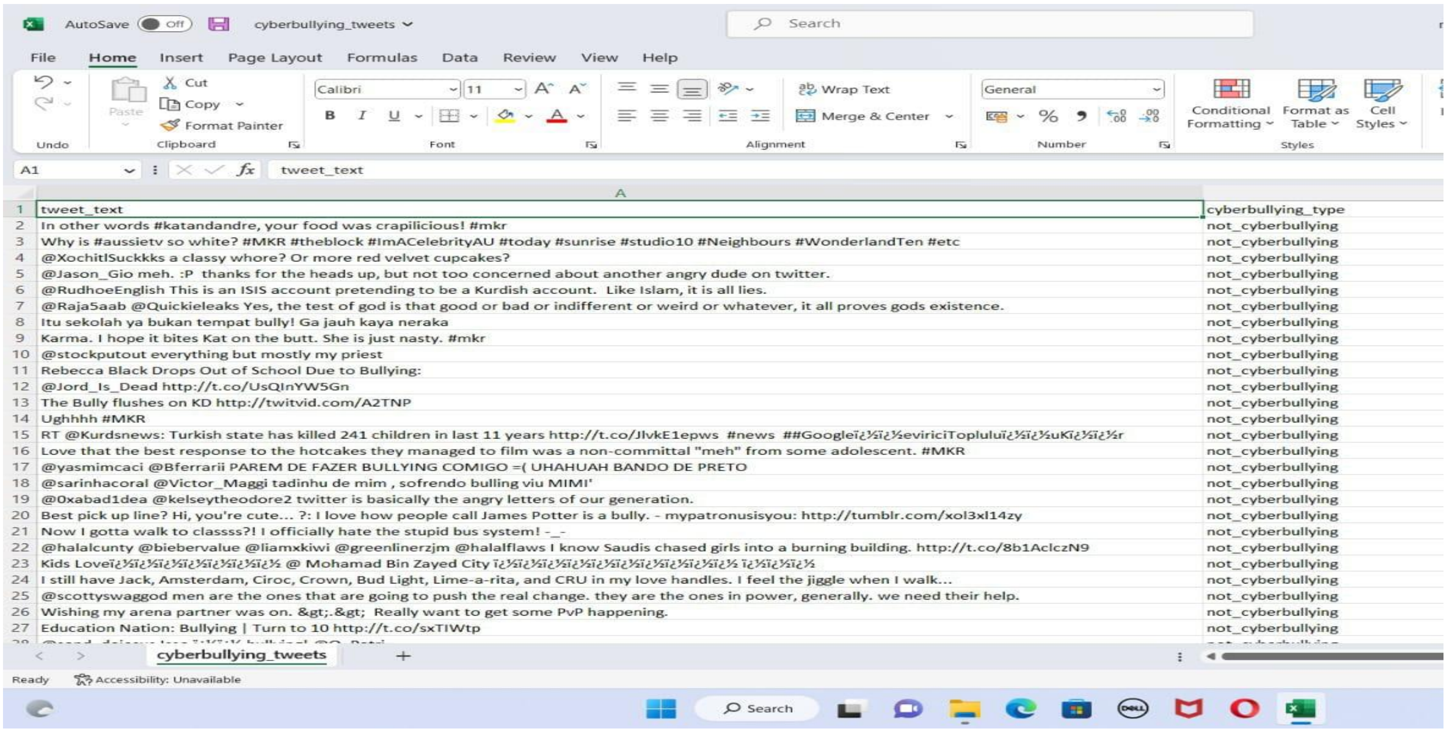
Please re- enter the new password

Reset Password

Made with Streamlit

Fig 6.3 Reset Password Page

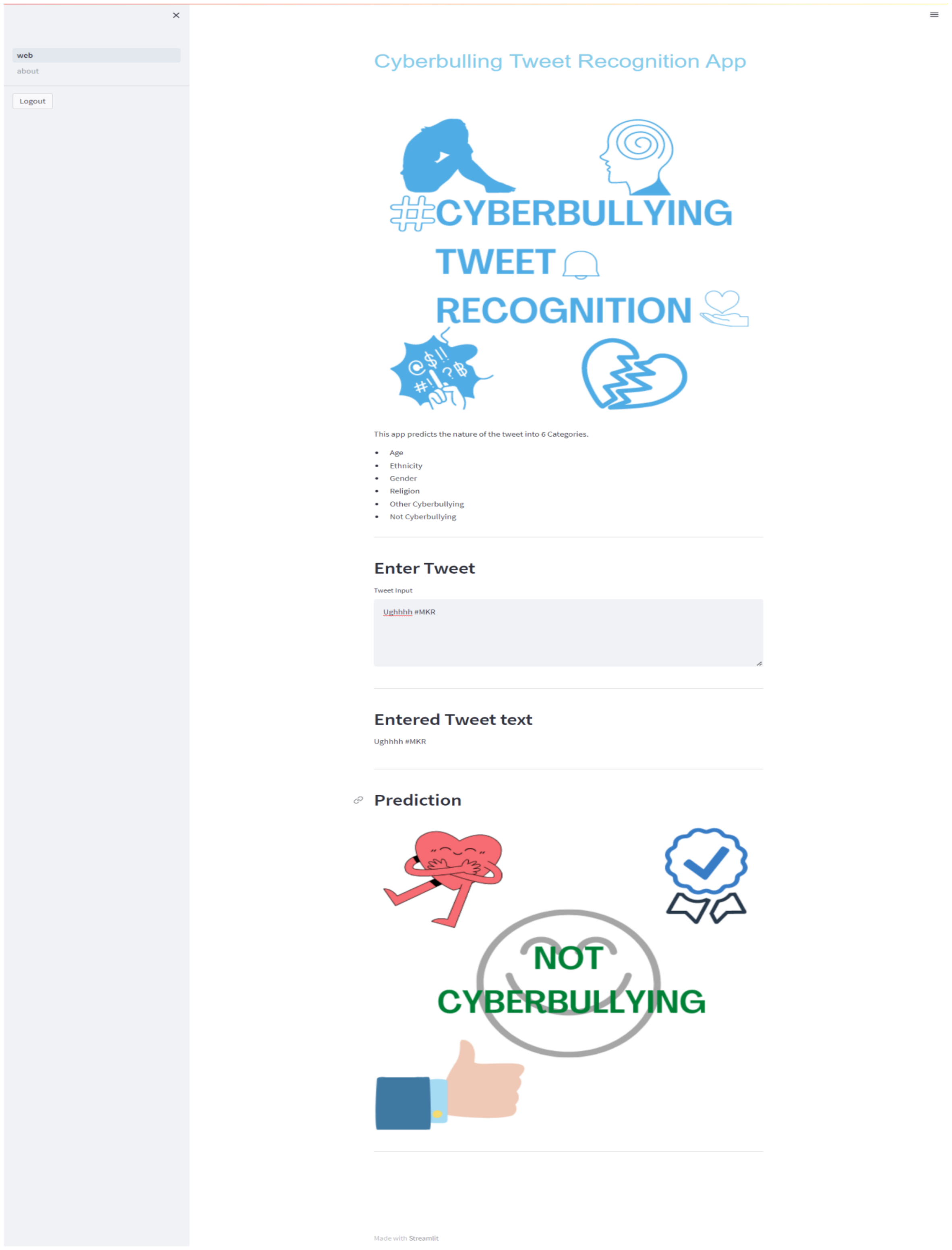
- **Excel Sheet**



	A	
1	tweet_text	cyberbullying_type
2	In other words #katandandre, your food was crapilicious! #mkr	not_cyberbullying
3	Why is #aussietv so white? #MKR #theblock #ImACelebrityAU #today #sunrise #studio10 #Neighbours #WonderlandTen #etc	not_cyberbullying
4	@XochitlSuckkks a classy whore? Or more red velvet cupcakes?	not_cyberbullying
5	@Jason_Gio meh. :P thanks for the heads up, but not too concerned about another angry dude on twitter.	not_cyberbullying
6	@RudhoeEnglish This is an ISIS account pretending to be a Kurdish account. Like Islam, it is all lies.	not_cyberbullying
7	@Raja5aab @Quickieleaks Yes, the test of god is that good or bad or indifferent or weird or whatever, it all proves gods existence.	not_cyberbullying
8	Itu sekolah ya bukan tempat bully! Ga jauh kaya neraka	not_cyberbullying
9	Karma. I hope it bites Kat on the butt. She is just nasty. #mkr	not_cyberbullying
10	@stockputout everything but mostly my priest	not_cyberbullying
11	Rebecca Black Drops Out of School Due to Bullying:	not_cyberbullying
12	@Jord_Is_Dead http://t.co/UsQInYW5Gn	not_cyberbullying
13	The Bully flushes on KD http://twitvid.com/A2TNP	not_cyberbullying
14	Ughhhh #MKR	not_cyberbullying
15	RT @Kurdsnews: Turkish state has killed 241 children in last 11 years http://t.co/JlvkE1epws #news ##Googleï¿½ï¿½eviriciTopluluï¿½ï¿½uKï¿½ï¿½r	not_cyberbullying
16	Love that the best response to the hotcakes they managed to film was a non-committal "meh" from some adolescent. #MKR	not_cyberbullying
17	@yasmimcaci @Bferrarii PAREM DE FAZER BULLYING COMIGO =(UHAHUAH BANDO DE PRETO	not_cyberbullying
18	@sarinhacoral @Victor_Maggi tadinhu de mim , sofrendo bulling viu MIMI'	not_cyberbullying
19	@0xabad1dea @kelseytheodore2 twitter is basically the angry letters of our generation.	not_cyberbullying
20	Best pick up line? Hi, you're cute... ?: I love how people call James Potter is a bully. - mypatronusisyou: http://tumblr.com/xol3xl14zy	not_cyberbullying
21	Now I gotta walk to classss?! I officially hate the stupid bus system! -_-	not_cyberbullying
22	@halalcunty @biebervalue @liamxkiwi @greenlinerzjm @halalflaws I know Saudis chased girls into a burning building. http://t.co/8b1AclczN9	not_cyberbullying
23	Kids Loveï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ @ Mohamad Bin Zayed City ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ï¿½ ï¿½ï¿½ï¿½	not_cyberbullying
24	I still have Jack, Amsterdam, Ciroc, Crown, Bud Light, Lime-a-rita, and CRU in my love handles. I feel the jiggle when I walk...	not_cyberbullying
25	@scottyswaggod men are the ones that are going to push the real change. they are the ones in power, generally. we need their help.	not_cyberbullying
26	Wishing my arena partner was on. &gt;.&gt; Really want to get some PvP happening.	not_cyberbullying
27	Education Nation: Bullying | Turn to 10 http://t.co/sxTIWtp	not_cyberbullying

Fig 6.4 Excel Sheet Page

- **Project Screen - 1**



Cyberbulling Tweet Recognition App

This app predicts the nature of the tweet into 6 Categories.

- Age
- Ethnicity
- Gender
- Religion
- Other Cyberbullying
- Not Cyberbullying

Enter Tweet

Tweet Input

Ughhhh #MKR

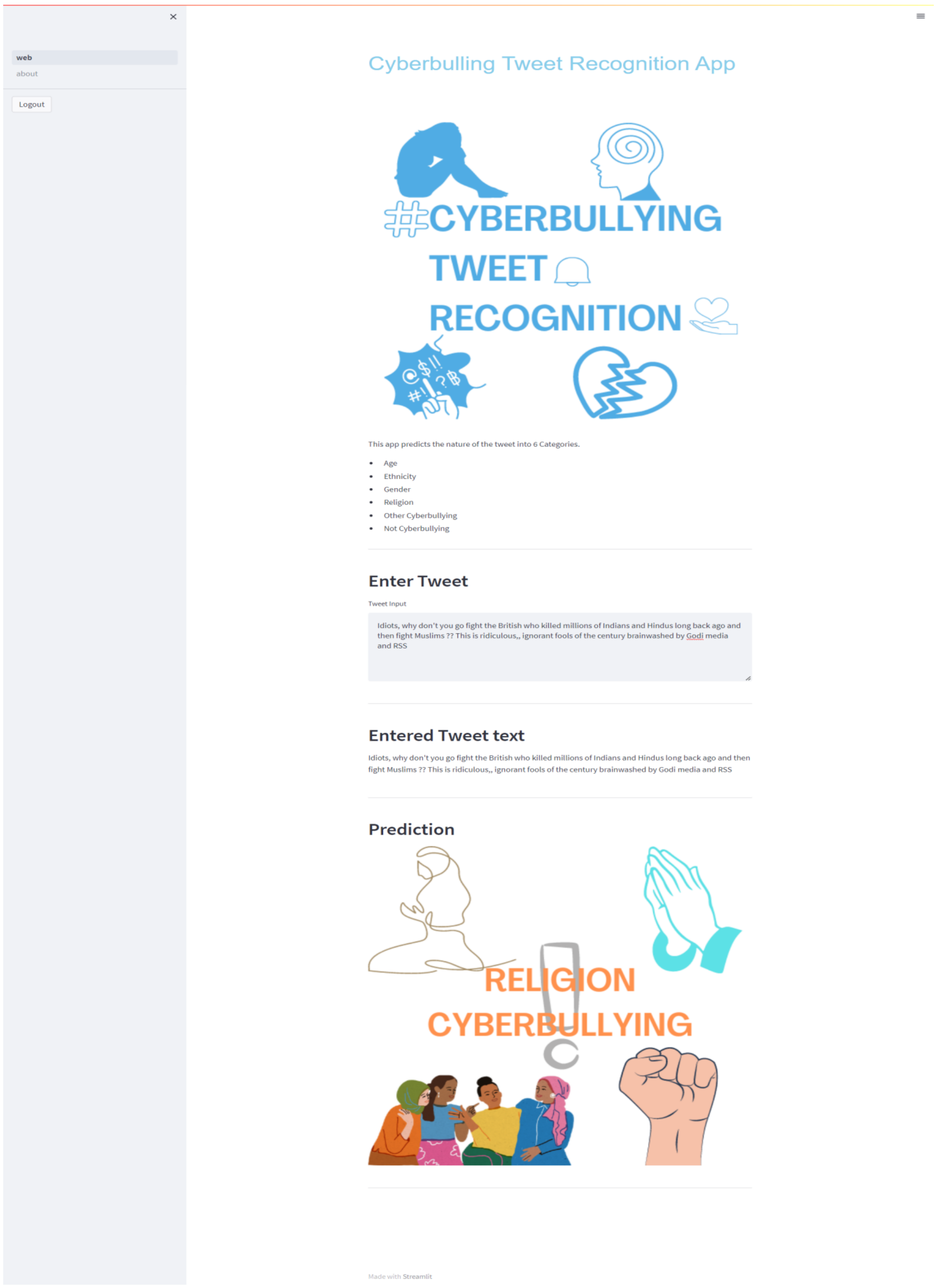
Entered Tweet text

Ughhhh #MKR

Prediction

Fig 6.5 Project Screen - 1

- **Project Screen - 2**



This app predicts the nature of the tweet into 6 Categories.

- Age
- Ethnicity
- Gender
- Religion
- Other Cyberbullying
- Not Cyberbullying

Enter Tweet

Tweet Input

Idiots, why don't you go fight the British who killed millions of Indians and Hindus long back ago and then fight Muslims ?? This is ridiculous,, ignorant fools of the century brainwashed by Godi media and RSS

Entered Tweet text

Idiots, why don't you go fight the British who killed millions of Indians and Hindus long back ago and then fight Muslims ?? This is ridiculous,, ignorant fools of the century brainwashed by Godi media and RSS

Prediction

Fig 6.6 Project Screen - 2

CHAPTER 7

SCHEDULE WORK

7.1 PROJECT MANAGEMENT

Month	**Week**	**Task**
August	1-2 3-4	Searching topics and also latest papers based on the topics. Searching three topics and also the latest papers based on website and applications.
September	1-2	We select 3 topics namely. 1.Smart e-health prediction system. 2.Farming assistance web portal. 3.Detection of Cyberbullying on social media using machine learning.
October	1-2	From above topics the topic third is selected i. e. “Detection of Cyberbullying on Social Media Using Machine Learning”
November	1-2	Then we collected more information about the topics

December	2-3	Then we collect more information about the Twitter
January	1-4	Then We collect more information about Machine Learning , Python (Jupyter Notebook , Numpy , Pandas , etc.)
February	2-4	Create dataset (Bullying or Non-Bullying)
March	1-4	Then we have to add six categories of prediction . 1. Age 2. Ethnicity 3. Gender 4. Religion 5. Other Cyberbulling 6. Not Cyberbulling
April	1-2	Successfully detect the Twitter comments are bullying or Non-Bullying

CHAPTER 8

CONCLUSION

In our proposed system we are able to implement our project in a fair and good manner with respect to the scope and relevance of social networking. In modern life the incidents of cyber bullying are very high. Every user of every age group is vulnerable to such threats. So, our project as a steady intension in irradiating such incidents through our social network. This is a social network which can be used by all age groups. Since this is very user friendly it is easily accessible by anyone and above all due to its relevance it is expected to be a success. The interface is very user friendly and hence it provides ease of usability.ss

This project also addresses the text-based cyberbullying detection problem, where robust and discriminative representations of messages are critical for an effective detection system. By designing semantic dropout noise and enforcing sparsity, we have developed semantic-enhanced marginalized denoising autoencoder as a specialized representation learning model for cyberbullying detection. In addition, word embeddings have been used to automatically expand and refine bullying word lists that is initialized by domain knowledge. The performance of our approaches has been experimentally verified through two cyberbullying corporation from social medias: Twitter and Myspace. As a next step we are planning to further improve the robustness of the learned representation by considering word order in messages.

CHAPTER 9

CODING

```
import streamlit as st

from streamlit_login_auth_ui.widgets import _login_


# importing libraries


from ctypes import alignment

from urllib import response

import pandas as pd

import streamlit as st

import altair as alt

from PIL import Image

import pandas as pd

import numpy as np

import re

import string

from nltk.stem import WordNetLemmatizer

from sklearn.model_selection import train_test_split

from sklearn.svm import SVC

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.preprocessing import LabelEncoder

from nltk.tokenize import RegexpTokenizer

from nltk import PorterStemmer, WordNetLemmatizer
from functions import *

import pickle


new_title = '<p style="font-family:sans-serif; color:skyblue; font-size: 42px;">Cyberbulling Tweet
Recognition App</p>'

st.markdown(new_title, unsafe_allow_html=True)



_loginobj = __login_(auth_token = "courier_auth_token",

                company_name = "Shims",

                width = 200, height = 250,

                logout_button_name = 'Logout', hide_menu_bool = False,

                hide_footer_bool = False,

                lottie_url = 'https://assets2.lottiefiles.com/packages/lf20_jcikwtux.json')



LOGGED_IN = _login_obj.build_login_ui()


if LOGGED_IN == True:

    image = Image.open('images/logo.png')


    st.image(image, use_column_width= True)


    st.write('''


  This app predicts the nature of the tweet into 6 Categories.
  * Age

  * Ethnicity

  * Gender

  * Religion

  * Other Cyberbullying

  * Not Cyberbullying


  *

  ''')


  # Text Box

    st.header('Enter Tweet ')

    tweet_input = st.text_area("Tweet Input", height= 150)

    print(tweet_input)

    st.write('''

    *

    ''')


    # print input on webpage

    st.header("Entered Tweet text ")

    if tweet_input:

      tweet_input

    else:

      st.write('''

      **No Tweet Text Entered!**

      ''')
```

```
st.write('''
*
''')


# Output on the page
st.header("Prediction")
if tweet_input:
    prediction = custom_input_prediction(tweet_input)
    if prediction == "Age":
        st.image("images/age_cyberbullying.png",use_column_width= True)
    elif prediction == "Ethnicity":
        st.image("images/ethnicity_cyberbullying.png",use_column_width= True)
    elif prediction == "Gender":
        st.image("images/gender_cyberbullying.png",use_column_width= True)
    elif prediction == "Not Cyberbullying":
        st.image("images/not_cyberbullying.png",use_column_width= True)
    elif prediction == "Other Cyberbullying":
        st.image("images/other_cyberbullying.png",use_column_width= True)
    elif prediction == "Religion":
        st.image("images/religion_cyberbullying.png",use_column_width= True)
else:
    st.write('''
    **No Tweet Text Entered!**
    ''')


st.write('''*''')
```

```
x`# importing libraries

import pandas as pd

import numpy as np

import re

import string

from nltk.stem import WordNetLemmatizer

from sklearn.model_selection import train_test_split

from sklearn.svm import SVC

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.preprocessing import LabelEncoder

from nltk.tokenize import RegexpTokenizer

from nltk import PorterStemmer, WordNetLemmatizer

import pickle

import nltk

nltk.download('wordnet')


# preprocessing functions


# converting tweet text to lower case

def text_lower(text):

    return text.str.lower()


# removing stopwoords from the tweet text

def clean_stopwords(text):

    # stopwords list that needs to be excluded from the data
    stopwordlist = ['a', 'about', 'above', 'after', 'again', 'ain', 'all', 'am', 'an',
             'and','any','are', 'as', 'at', 'be', 'because', 'been', 'before',
             'being', 'below', 'between','both', 'by', 'can', 'd', 'did', 'do',
             'does', 'doing', 'down', 'during', 'each','few', 'for', 'from',
             'further', 'had', 'has', 'have', 'having', 'he', 'her', 'here',
             'hers', 'herself', 'him', 'himself', 'his', 'how', 'i', 'if', 'in',
             'into','is', 'it', 'its', 'itself', 'just', 'll', 'm', 'ma',
             'me', 'more', 'most','my', 'myself', 'now', 'o', 'of', 'on', 'once',
             'only', 'or', 'other', 'our', 'ours','ourselves', 'out', 'own', 're',
             's', 'same', 'she', "shes", 'should', "shouldve",'so', 'some', 'such',
             't', 'than', 'that', "thatll", 'the', 'their', 'theirs', 'them',
             'themselves', 'then', 'there', 'these', 'they', 'this', 'those',
             'through', 'to', 'too','under', 'until', 'up', 've', 'very', 'was',
             'we', 'were', 'what', 'when', 'where','which','while', 'who', 'whom',
             'why', 'will', 'with', 'won', 'y', 'you', "youd","youll", "youre",
             "youve", 'your', 'yours', 'yourself', 'yourselves']
    STOPWORDS = set(stopwordlist)
    return " ".join([word for word in str(text).split() if word not in STOPWORDS])


# cleaning and removing punctuations
def clean_puctuations(text):
    english_puctuations = string.punctuation
    translator = str.maketrans('','', english_puctuations)
    return text.translate(translator)


# cleaning and removing repeating characters
```

```
def clean_repeating_characters(text):
    return re.sub(r'(.)1+', r'1', text)


# cleaning and removing URLs
def clean_URLs(text):
    return re.sub(r"((www.[^s]+)|(http\S+))","",text)


# cleaning and removing numeric data
def clean_numeric(text):
    return re.sub('[0-9]+', '', text)


# Tokenization of tweet text
def tokenize_tweet(text):
    tokenizer = RegexpTokenizer('\w+')
    text = text.apply(tokenizer.tokenize)
    return text


# stemming
def text_stemming(text):
    st = PorterStemmer()
    text = [st.stem(word) for word in text]
    return text


# lemmatization
def text_lemmatization(text):
    lm = WordNetLemmatizer()
    text = [lm.lemmatize(word) for word in text]
    return text


# defining preprocess function


def preprocess(text):
    text = text_lower(text)
    text = text.apply(lambda text: clean_stopwords(text))
    text = text.apply(lambda x : clean_puctuations(x))
    text = text.apply(lambda x: clean_repeating_characters(x))
    text = text.apply(lambda x : clean_URLs(x))
    text = text.apply(lambda x: clean_numeric(x))
    text = tokenize_tweet(text)
    text = text.apply(lambda x: text_stemming(x))
    text = text.apply(lambda x: text_lemmatization(x))
    text = text.apply(lambda x : " ".join(x))
    return text


# Function for custom input prediction
def custom_input_prediction(text):
    import nltk
    nltk.download('omw-1.4')
    text = pd.Series(text)
    text = preprocess(text)
    text = [text[0],]
    # to use this function we will need to define vectoriser first
```

```
vectoriser = pickle.load(open("tdf_vectorizer", "rb"))

text = vectoriser.transform(text)

model = pickle.load(open("model.bin", "rb"))

prediction = model.predict(text)

prediction = prediction[0]


interpretations = {

    0 : "Age",

    1 : "Ethnicity",

    2 : "Gender",

    3 : "Not Cyberbullying",

    4 : "Other Cyberbullying",

    5 : "Religion"

}


for i in interpretations.keys():

    if i == prediction:

        return interpretations[i]
```

BIBILIOGRAPHY

- Niloofar Safi Samghabadi , a Pastor L' opez-Monroy, Thamar Solorio, Marseille, 11-16 May 2020

- http://www.pcmag.com/artical2/0,2817,2388540,00.asp

- Aditya Desai1, Shaahank Kalaskar2, Omkar Kumbhar3, and rashmi Dhuma14, https://doi.org/10.1051/itmconf/20214003038 ITM Web of Conferences 40, 03038(2021)

- Miss. Jafri Sayeedaaliza Abutorab*1, Miss. Wagh Roshani Balasaheb*2, Miss. Gaikwad Vaishnavi Subodh*3, Miss. Sonawane Ujjwala Dattu*4, Professor Waghmare. A.I. *504/Issue:05/May-2022