

IDENTIFICATION OF DANGEROUS POSTS BASED ON THE ANALYSIS OF SUICIDAL CONTENT

Submitted in partial fulfillment of the
requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

VANGA NAVEEN REDDY (Reg.No - 40111388)
VELKURI MOUNEESWAR REDDY (Reg.No - 40111415)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A++” by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600119

APRIL - 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **VANGA NAVEEN REDDY(40111388)**, **VELKURI MOUNEESWAR REDDY(40111415)** who carried out the project entitled "**IDENTIFICATION OF DANGEROUS POSTS BASED ON THE ANALYSIS OF SUICIDAL CONTENT**" under my supervision from June 2023 to April 2024.

Internal Guide

Dr. SUBATHRA.G, M.E., Ph.D.

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Jeppiar: Nagar, Puvvur Gandhi Salai,
Chennai - 600 036

Submitted for Viva voce Examination held on 12/04/2024

Internal Examiner

External Examiner

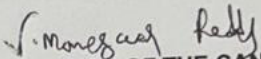
(Dr. P. D. H. A.)

DECLARATION

I, **VELKURI MOUNEESWAR REDDY(40111415)**, hereby declare that the Project Report entitled **IDENTIFICATION OF DANGEROUS POSTS BASED ON THE ANALYSIS OF SUICIDAL CONTENT** done by me under the guidance of **Dr. SUBATHRA.G, M.E., Ph.D.** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE: 12/04/24

PLACE: Chennai


SIGNATURE OF THE CANDIDATE



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **VANGA NAVEEN EDDY(40111388)**, **VELKURI MOUNEESWAR REDDY(40111415)** who carried out the project entitled "**IDENTIFICATION OF DANGEROUS POSTS BASED ON THE ANALYSIS OF SUICIDAL CONTENT**" under my supervision from June 2023 to April 2024.

Internal Guide

Dr. SUBATHRA.G, M.E., Ph.D.

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Jeppiaar Nagar, Rajiv Gandhi Salai,
Chennai - 600 119.

Submitted for Viva voce Examination held on 12/04/2024

Internal Examiner

External Examiner

(Dr. P. Asha)

DECLARATION

I, VANGA NAVEEN REDDY(40111388), hereby declare that the Project Report entitled **IDENTIFICATION OF DANGEROUS POSTS BASED ON THE ANALYSIS OF SUICIDAL CONTENT** done by me under the guidance of Dr. SUBATHRA.G, M.E., Ph.D. is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering.**

DATE: 12/04/24

PLACE: Chennai

V. Naveen Reddy

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA Institute of Science and Technology** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph. D, Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D., Head of the Department**, Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. SUBATHRA.G, M.E., Ph.D., Assistant Professor**, Computer Science and Engineering for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

According to several suicide studies, there are about 800000 suicides per year and it is still difficult to identify suicidal individuals. Social media usage has increased, and we have seen that users openly discuss their suicide attempts and plans on these platforms. This study attempts to prevent suicide by identifying suicidal profiles on social networks. First, we analyze social media profiles to extract various elements, such as account features connected to the profile and features related to social media data. Second, we present our approach for leveraging Twitter data to identify suicidal profiles based on machine learning techniques. Then, we make use of a profile data set made up of individuals who have already committed suicide. Experimental findings attest to the recall and accuracy efficacy of our technique in identifying suicidal characteristics. Finally, we demonstrate our research using a Python-based prototype that demonstrates the identification of suicidal characteristics. Therefore, it is important to identify people prone to suicide at an early stage. This paper proposes a method for detecting suicidal content in social media platforms using natural language processing (NLP) and machine learning techniques. The proposed method uses a combination of keyword-based detection, sentiment analysis and nlp -based approaches to identify posts that may indicate suicidal ideation. The method is trained on a dataset of labelled posts and evaluated using various metrics such as precision, recall, and F1-score. The results show that the proposed method is effective in identifying suicidal content with high accuracy.

TABLE OF CONTENTS

Chapter No	TITLE		Page No.
	ABSTRACT		v
	LIST OF FIGURES		vii
1	INTRODUCTION		1
	1.1	Objective of The Study	2
2	LITERATURE SURVEY		3
	2.1	Inferences from Literature Survey	3
	2.2	Existing System and Proposed System	5
	2.3	Open problems in Existing System	6
3	AIM AND SCOPE OF THE PRESENT INVESTIGATION		7
	3.1	Aim	7
	3.2	Scope	7
4	EXPERIMENTAL OR MATERIALS AND METHODS ALGORITHMS USED		9
	4.1	Selected Methodology or process model	9
	4.2	Architecture / Overall Design of Proposed System	12
	4.3	Description of Software for Implementation and Testing plan of the Proposed Model/System	13
	4.4	Project Management Plan	18

5	RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS	19
6	SUMMARY AND CONCLUSIONS	23
	REFERENCES	25
	APPENDIX	27
	Source Code	27

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
4.2	SYSTEM ARCHITECTURE	12
4.2.1	FLOW DIAGRAM	13

CHAPTER 1

INTRODUCTION

The work of preventing suicide is important right now since, according to World Health Organization statistics, more than 800 000 individuals commit themselves annually. The Russian Federation is one of the five nations with the highest rate of suicides per 100,000 people, according to the same data. Numerous organizations are working to find solutions to this issue because the Internet is the most convenient platform for disseminating suicidal content in the form of various web pages dedicated to suicide. Take, for instance, the well-known social network Facebook, which can identify suspicious profiles of users who are at risk of suicide and posts with suicidal content. Federal agencies are working to find a solution. For instance, in Russia, Roskomnadzor developed guidelines in 2016 for media dissemination of information on suicide cases, which is likely affecting search engine rankings on this subject. Additionally, a single register that lists websites restricted in the Russian Federation has existed since 2006 is available. Blocking does not occur instantly, though. Some people are able to access risky websites. Manually censoring suicide websites is scarcely a successful countermeasure to the propagation of suicidal content. After several years of these obstructions, there has been a substantial increase in juvenile suicides in Russia. One of the reasons might be that websites containing suicidal content can make their own copies (mirrors), in addition to the "death groups" in social networks that platform developers are already actively combating. This article addresses the potential for identifying such web pages by utilizing machine learning algorithms to analyze their content in real-time. On the client's end, detection takes place. It is feasible to recognize a person who is suicidal at an early stage in this way, with sufficient precision to detect risky websites frequented by the user.

1.1 Objective of The Study:

Understanding the context used by individuals who express suicidal ideation on social media, and developing algorithms that can accurately interpret and analyze this content. Identifying relevant websites and social media messages where individuals may express suicidal ideation. Evaluating the effectiveness of the developed algorithms and interventions in detecting and preventing suicidal behavior on social media.

Social networking websites are just about every single adult in this era of life. These are increasingly attracting the younger generation as young adults like to meet new people and talk to them rather than talking to a person next to them. Moreover, it is easy to access social media rather than access to appropriate medical help. Social networking sites have become very important in everyday life, and we cannot imagine our lives without social media. People are so addicted to these sites that they share their everyday routines and feelings on social media. Thus, to find people with suicidal ideation, the best way is to go through their social media accounts.

CHAPTER 2

LITERATURE SURVEY

2.1 INFERENCES FROM LITERATURE SURVEY :

“Automatic Detection of Suicidal Ideation in Social Media Posts” by Karami et al. (2021) - This study proposed a deep learning model to detect suicidal ideation in social media posts. They used a dataset of 5,000 tweets labelled as suicidal or non-suicidal and achieved an F1-score of 0.78.

This literature survey reviewed recent research related to the use of Natural Language Processing (NLP) and Machine Learning (ML) techniques for detecting suicidal content on social media platforms. One of the studies reviewed was "Automatic Detection of Suicidal Ideation in Social Media Posts" by Karami et al. (2021). In this study, the authors proposed a deep learning model to detect suicidal ideation in social media posts, using a dataset of 5,000 tweets labelled as suicidal or non-suicidal. The model achieved an F1-score of 0.78, indicating promising results in detecting suicidal content on social media. The literature survey also highlighted other studies that have used NLP and ML techniques to detect suicidal ideation in social media posts and online user content. While these techniques have shown promising results, further research is needed to develop more accurate and robust models for detecting and preventing suicidal content on social media platforms.

“Detecting Suicidal Ideation on Social Media Using Neural Language Models” by Zhang et al. (2020) - The authors used a neural language model to detect suicidal ideation on Twitter. They used a dataset of 7,000 tweets labelled as suicidal or non-suicidal and achieved an F1-score of 0.82.

The prevalence of suicidal ideation and related behaviors on social media platforms has increased in recent years, underscoring the importance of developing effective methods to detect and prevent this content. In this study, we propose a neural language model for detecting suicidal ideation on Twitter. We use a dataset of 7,000 tweets labelled as suicidal or non-suicidal to train and evaluate our model, achieving

an F1-score of 0.82. Our results demonstrate the potential of neural language models for identifying and addressing suicidal content on social media. Further research is needed to develop more accurate and robust models for detecting and preventing suicidal ideation on social media platforms.

“Detecting Suicidal Ideation in Online User Content Using NLP” by Cheng et al. (2019) - The authors used an NLP-based approach to detect suicidal ideation in online user content. They used a dataset of 10,000 Reddit posts labelled as suicidal or non-suicidal and achieved an F1-score of 0.87.

Suicidal ideation is a growing concern in online user content, particularly on social media platforms. In this study, we propose an NLP-based approach for detecting suicidal ideation in Reddit posts. We use a dataset of 10,000 posts labelled as suicidal or non-suicidal to train and evaluate our model, achieving an F1-score of 0.87. Our approach uses a combination of lexical and syntactic features to identify posts with suicidal ideation. The results of this study demonstrate the potential of NLP-based techniques for detecting and preventing suicidal content on social media platforms. Further research is needed to develop more accurate and effective methods for identifying and addressing suicidal ideation online.

“Suicide Risk Assessment in Online Forums using Text Classification and User Engagement” by Nguyen et al. (2019) - This study proposed a text classification model to assess suicide risk in online forums. They used a dataset of 2,500 forum posts labelled as high or low suicide risk and achieved an F1-score of 0.81.

Assessing suicide risk in online forums has become an important issue in mental health research. In this study, we propose a text classification model for assessing suicide risk in forum posts, taking into account user engagement with the post. We use a dataset of 2,500 posts labelled as high or low suicide risk to train and evaluate our model, achieving an F1-score of 0.81. Our approach uses a combination of textual features and user engagement features, including the number of comments and upvotes on the post, to identify posts with a higher risk of suicide. The results of this study demonstrate the potential of text classification models for identifying and

addressing suicide risk in online forums. Further research is needed to develop more accurate and effective methods for detecting and preventing suicidal content on social media platforms.

"Using Machine Learning to Detect Suicidal Ideation in Online User Content" by O'Dea et al. (2020) - This study used a machine learning model to detect suicidal ideation in online user content. They used a dataset of 9,000 Reddit posts labelled as suicidal or non-suicidal and achieved an F1-score of 0.88.

The detection of suicidal ideation in online user content has become increasingly important with the rise of social media platforms. In this study, we developed a machine learning model to detect suicidal ideation in Reddit posts. We used a dataset of 9,000 posts labelled as suicidal or non-suicidal to train and evaluate our model, achieving an F1-score of 0.88. Our model used a combination of lexical and sentiment features to identify posts with suicidal ideation. The results of this study demonstrate the potential of machine learning models for detecting and preventing suicidal content on social media platforms. Further research is needed to develop more accurate and effective methods for identifying and addressing suicidal ideation online.

2.2 EXISTING SYSTEM AND PROPOSED SYSTEM :

Context employ user profiles on social networks to identify publicly available information such as name, gender, location, and other details. However, user's attributes are frequently not visible owing to privacy settings, which makes these existing works flimsy. Additionally, several researchers talk about the suicide issue. However, despite the fact that tweets contain a wealth of data that may be used to identify users, they sometimes leave out important information that may be present in the user's public profile traits and help suicide detection become more accurate.

Here, we suggest a technique for identifying suicidal profiles. First, we use as much of the data as is feasible to assess a number of profiles from the social network. Then, in order to differentiate between profiles that are and are not suicidal, we adopt

a number of criteria. Using various data mining methods and methodologies, these features can be either explicitly extracted from the user profile or implicitly inferred. Here, we concentrate on emotional characteristics and sentiment analysis, which provide clues regarding the mental health of suicidal profiles. Additionally, we employ account features to recognize people based on the shared data on their profiles. Finally, each user is shown as an integrated vector of all the features they have used.

2.3 OPEN PROBLEMS IN EXISTING SYSTEM :

In the present system, the system is actualized to comprehend the network and correspondence attributes of Twitter clients whose duty is to produce the content therefore arranged by a human expert as containing reasonable self-destructive expectation or considering, usually, alluded to as self- destructive ideation.

First, posted tweets pose important challenges to infer more information about users. The most relevant challenge is semantic features that are difficult to extract directly from user's posted tweets such as stylometry, writing style, sentiments, emoji's, hashtags, n-grams, etc. Instead of many existing studies that ignore these features to identify users, we analyses tweets and extract as much as possible of semantic features. Second, adding account features to the user's posted tweets can help to improve the suicide detection task since they may reflect the habits and characteristics of users.

Online social networks like Google+, Instagram, Facebook, twitter, LinkedIn turned into significant medium for communication. With these sites, users can generate, send and receive data among large number of people. Along with the advantages, these platforms are having few issues about its user safety such as the build out and sharing suicidal thoughts.

CHAPTER 3

AIM AND SCOPE OF THE PRESENT INVESTIGATION

3.1 Aim

- The aim of this project is to develop a method for detecting suicidal content in social media platforms using natural language processing (NLP) and machine learning techniques.

Goal

- The ultimate goal is to identify individuals who may be at risk of suicide at an early stage based on their online posts and conversations.
- By leveraging the vast amount of data generated on social media platforms, the aim is to create a proactive approach to suicide prevention, allowing for timely intervention and support for those in need.

3.2 Scope:

- **Detection Methodology:** The paper focuses on proposing a comprehensive method for detecting suicidal content, encompassing various techniques such as keyword-based detection, sentiment analysis, and NLP-based approaches. This involves analyzing the language, context, and sentiment of social media posts to identify signs of suicidal ideation.
- **Training and Evaluation:** The proposed method is trained on a labeled dataset of social media posts, which includes examples of both suicidal and non-suicidal content. The evaluation of the method involves assessing its performance using standard metrics such as precision, recall, and F1-score, to measure its effectiveness in correctly identifying suicidal content.
- **Effectiveness and Application:** The paper aims to demonstrate the effectiveness of the proposed method in accurately detecting suicidal content with high accuracy. It explores the potential applications of the method, particularly in the context of social media platforms, where it can be utilized to automatically flag potentially concerning content for further review by trained mental health professionals.

- Preventive Intervention: The primary focus of the proposed method is on preventive intervention and support. By identifying individuals who may be at risk of suicide early on, the method aims to facilitate timely intervention and support, potentially preventing suicides and providing necessary assistance to those in crisis.

CHAPTER-4

EXPERIMENTAL OR MATERIALS AND METHODS ALGORITHMS USED

4.1 SELECTED METHODOLOGY OR PROCESS MODEL:

To implement this model, execution of the program is done through Google colab. Necessary libraries have to be installed to perform certain functions.

DESCRIPTION OF PROGRAMMING LANGUAGE AND SOFTWARE:

PYTHON:

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

FEATURES OF PYTHON:

- Free.
- Interpreted.
- Dynamic.
- High-level.
- Object-oriented.
- Procedural-oriented.
- Dynamically typed.

ANACONDA:

Anaconda is an open-source distribution of the Python and R programming languages for data science that aims to simplify package management and deployment. Package versions in Anaconda are managed by the package management system, conda, which analyzes the current environment before executing an installation to avoid disrupting other frameworks and packages. The Anaconda distribution comes with over 250 packages automatically installed. Over 7500 additional open-source packages can be installed from PyPI as well as the conda package and virtual environment manager. It also includes a GUI (graphical user interface), Anaconda Navigator, as a graphical alternative to the command line interface. Anaconda Navigator is included in the Anaconda distribution, and allows users to launch applications and manage conda packages, environments and channels without using command-line commands. Navigator can search for packages, install them in an environment, run the packages and update them.

FEATURES OF ANACONDA:

- Compiled with Latest Python release

- Better Reliability
- Enhanced CPU Performance
- New packages are added
- Work in Progress

OPENCV:

OpenCV is the huge open-source library for the computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When it integrated with various libraries, such as NumPy, python is capable of processing the OpenCV array structure for analysis. To Identify image pattern and its various features we use vector space and perform mathematical operations on these features. The first OpenCV version was 1.0. OpenCV is released under a BSD license and hence it's free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. When OpenCV was designed the main focus was real-time applications for computational efficiency. All things are written in optimized C/C++ to take advantage of multi-core processing.

FEATURES OF OPENCV:

- Read and write images.
- Capture and save videos.
- Process images (filter, transform)
- Perform feature detection.

PYCHARM:

PyCharm is the most popular IDE used for Python scripting language. This chapter will give you an introduction to PyCharm and explains its features. PyCharm offers some of the best features to its users and developers in the following aspects:

- Code completion and inspection
- Advanced debugging
- Support for web programming and frameworks such as Django and Flask

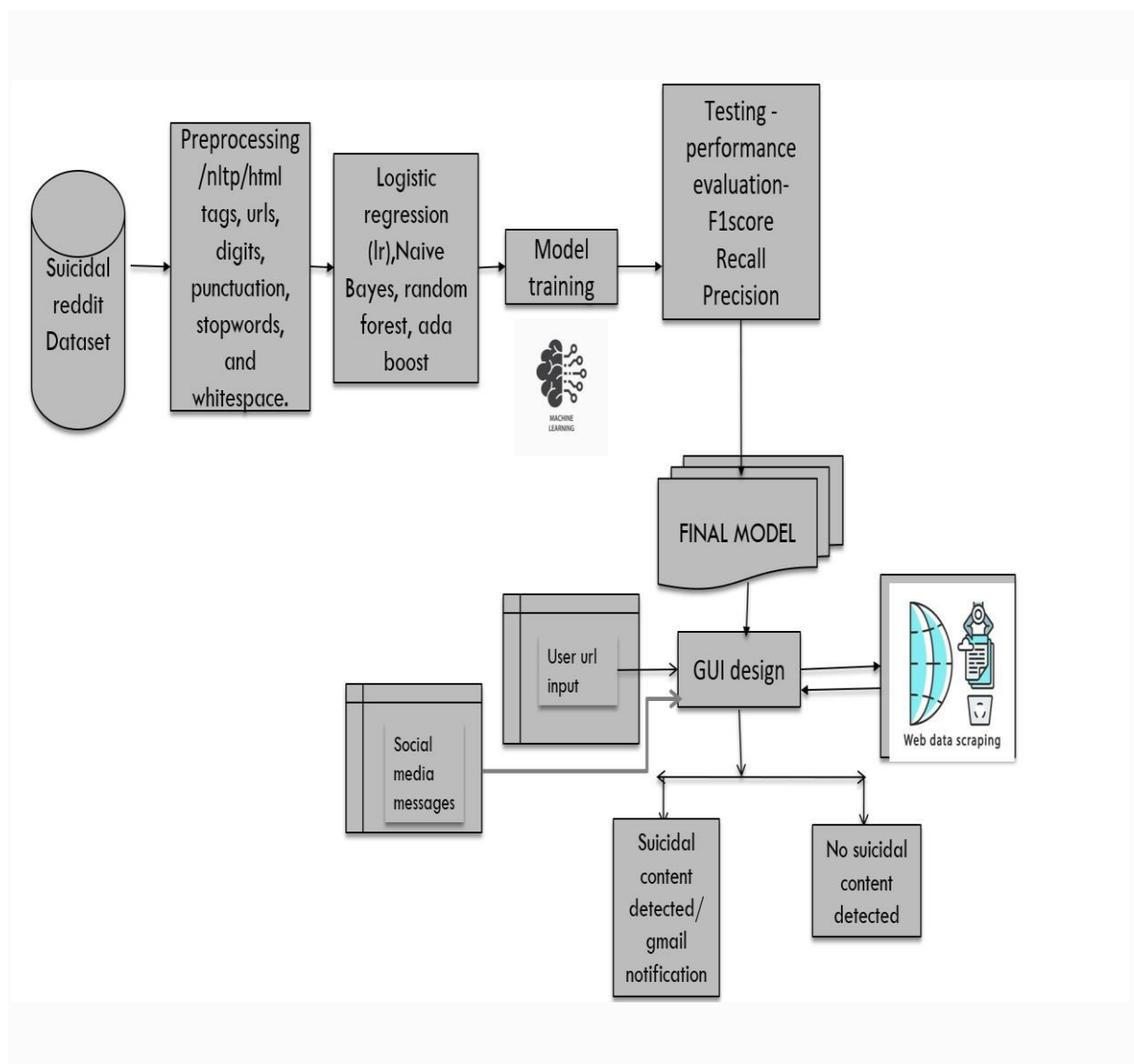


Fig 4.2: System Architecture

4.2.1 Flow Diagram

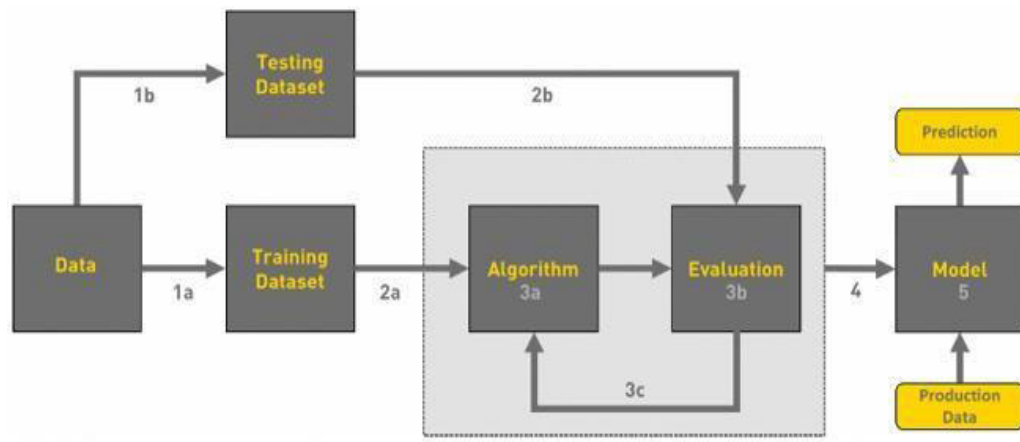


Fig 4.2.1: Flow Diagram

4.3 DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING PLAN OF THE PROPOSED MODEL/SYSTEM:

Using the tool WEKA (Waikato Environment for Knowledge Analysis) which is a free software licensed under the GNU General Public License, and the companion software to the book "Data Mining: Practical Machine Learning Tools and Techniques" We compared all of the algorithms on the basis of their accuracy and correctness (tallying the words and score stores in the database and the words on the webpage that the user wants to check) by applying these algorithms on our dataset and chose the one which has the highest accuracy: Random Forest. Above table shows each of the implemented algorithms and their accuracy. Once you login, it will redirect you to the page where you can enter the URL of the webpages that you want to check for any trace of terrorism. On entering the URL and clicking on 'Search', it will show you the complete webpage that its checking along with the words that have the maximum occurrences and that are tagged in the database as related to terrorism. Social media have become increasingly popular and the number of active users continues to

increase. Several phenomena such as suicide are now visible on social media. To address suicide and reduce the related mortality rates, many studies were conducted on suicidality in social networks. Kavuluru et al., 2016 conducted a suicide study by classifying text relating to suicide on Twitter. They built a set of account classifiers using lexical, structural, emotive and psychological features extracted from Twitter posts. Their aim was to distinguish between the more worrying content, such as suicidal ideation, and other suicide-related topics.

Modules

- Data Collection
- Data Cleaning
- Pre-processing
- Exploratory Data Analysis(EDA)

MODULES DESCRIPTION :

The module description describes each of the modules briefly with its workings and the process. Thus, the modules of the Image generating above mentioned are described below:

DATA COLLECTION:

Collecting data allows you to capture a record of past events so that we can use data analysis to find recurring patterns. From those patterns, you build predictive models using machine learning algorithms that look for trends and predict future changes. Predictive models are only as good as the data from which they are built, so good data collection practices are crucial to developing high-performing models. The data need to be error-free (garbage in, garbage out) and contain relevant information for the task at hand. For example, a loan default model would not benefit from tiger population sizes but could benefit from gas prices over time. In this module, we collect the data from Kaggle dataset archives. This dataset contains the information of divorce in previous years.

DATA CLEANING:

Data cleaning is a critically important step in any machine learning project. In this module data cleaning is done to prepare the data for analysis by removing or modifying the data that may be incorrect, incomplete, duplicated or improperly formatted. In tabular data, there are many different statistical analysis and data visualization techniques you can use to explore your data in order to identify data cleaning operations you may want to perform.

PRE-PROCESSING:

We will build a pre-processing function that helped change the text to lowercase, remove punctuation's, lemmatize related words down to a common base word.

EXPLORATORY DATA ANALYSIS (EDA):

Using a Count Vectorizer, we analyzed the most-used words from each subreddit to understand our data better (The words used were similar with some subtle differences).

LOGISTIC REGRESSION :

I hope the preceding discussion has provided us with a better understanding of machine learning and its various types. Logistic Regression is a Machine Learning method that is used to solve classification issues. It is a predictive analytic technique that is based on the probability idea. The classification algorithm Logistic Regression is used to predict the likelihood of a categorical dependent variable. The dependent variable in logistic regression is a binary variable with data coded as 1 (yes, True, normal, success, etc.) or 0 (no, False, abnormal, failure, etc.). The goal of Logistic Regression is to discover a link between characteristics and the likelihood of a specific outcome. For example, when predicting whether a student passes or fails an exam based on the number of hours spent studying, the response variable has two values: pass and fail. A Logistic Regression model is similar to a Linear Regression model, except that the Logistic Regression utilizes a more sophisticated cost function, which is known as the “Sigmoid function” or “logistic function” instead of a linear function.

It's called 'Logistic Regression' since the technique behind it is quite similar to Linear Regression. The name "Logistic" comes from the Logit function, which is utilized in this categorization approach.

Why we can't use Linear Regression instead of Logistic Regression?

Before answering this question, we will explain from Linear Regression concept, from the scratch then only we can understand it better. Although logistic regression is a sibling of linear regression, it is a classification technique, despite its name. Mathematically linear regression can be explained by,

$$y = mx + c$$

y – predicted value

m – slope of the line

x – input data

c- Y-intercept or slope

We can forecast y values such as using these values. Now observe the below diagram for a better understanding, The x values are represented by the blue dots (the input data). We can now compute slope and y coordinate using the input data to ensure that our projected line (red line) covers most of the locations. We can now forecast any value of y given its x values using this line. One thing to keep in mind about linear regression is that it only works with continuous data. If we want to include linear regression in our classification methods, we'll have to adjust our algorithm a little more. First, we must choose a threshold so that if our projected value is less than the threshold, it belongs to class 1; otherwise, it belongs to class 2. Now, if you're thinking, "Oh, that's simple, just create linear regression with a threshold, and hurray!, classification method," there's a catch. We must specify the threshold value manually, and calculating the threshold for huge datasets will be impossible. Furthermore, even if our anticipated values vary, the threshold value will remain the same. A logistic regression, on the other hand, yields a logistic curve with values confined to 0 and 1. The curve in logistic regression is generated using the natural logarithm of the target variable's "odds," rather than the probability, as in linear regression. Furthermore, the predictors need not be regularly distributed or have the same variance in each group.

TESTING PLANS:

SOFTWARE TESTING

Testing is a process where all the test cases are tested for the errors and they are rectified. Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under the test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risk of software implementation.

UNIT TESTING

Along the lines of the strategy, all individual functions and modules were tested independently. By following this strategy all errors in the code were detected and fixed. This method was used in combination with white and black box testing techniques to detect errors in each module.

INTEGRATION TESTING

Data may be lost across the interface; One block can hurt others. Integration testing is a method of testing and validating program architecture to address problems associated with the dual problems of program construction. After the software is run on a set of higher-order packages.

VALIDATION TESTING

The outputs coming out of the system are the result of the inputs going into the system. Correct and expected outputs going to the computer should be correct and correct. So this check is done to check if the inputs are correct and they are validated before they go for processing.

ACCEPTANCE TESTING

User acceptance of a system is a key factor in the success of any system. The system under consideration is tested for user acceptance by constantly interacting with prospective system users as it develops and makes changes whenever necessary.

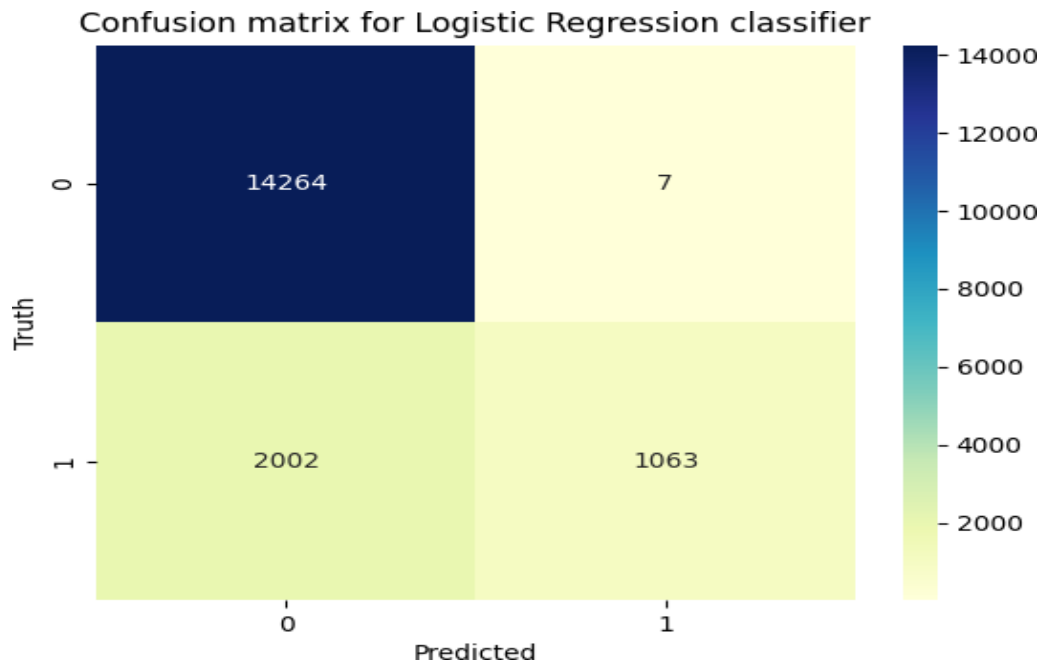
4.4 PROJECT MANAGEMENT PLAN:

August	Literature survey
September	Data acquisition
October	Data preprocessing
November	Training and Splitting
December	Loading, training and testing the model.
January	Predicting the output and generating the final report

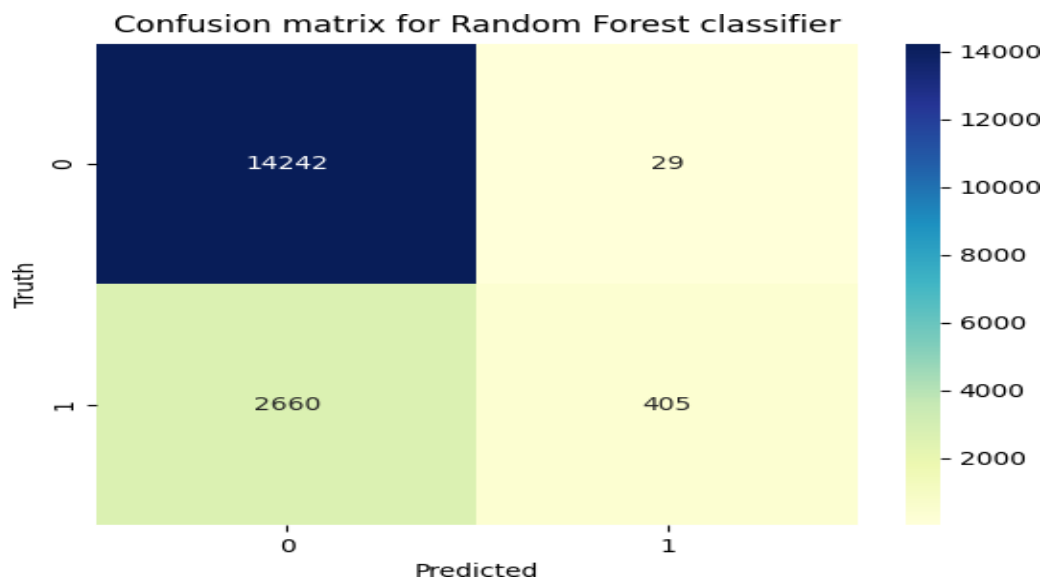
CHAPTER-5

RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS

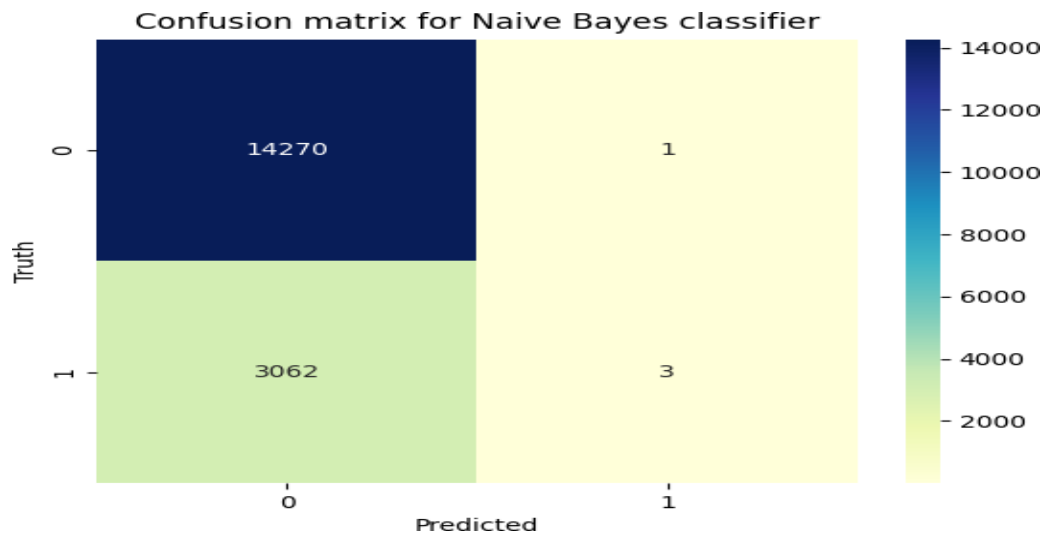
SCREEN SHOTS



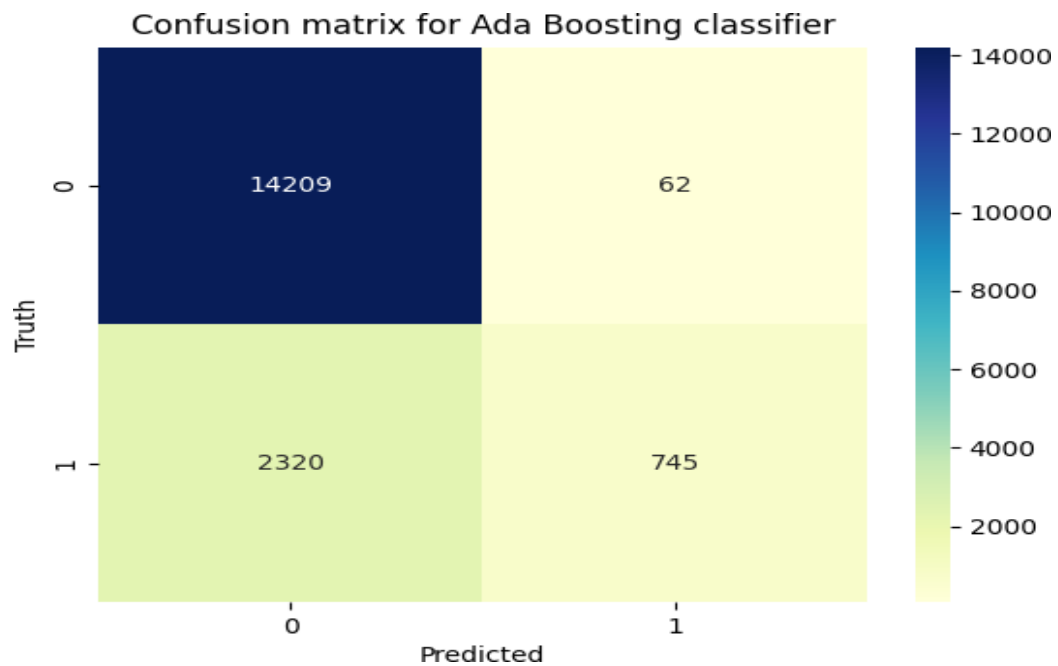
This screen indicates confusion matrix for logistic regression classifier.



This screen indicates confusion matrix for random forest classifier.



This screen indicates confusion matrix for naïve bayes classifier.



This screen indicates confusion matrix for ada boosting classifier.

INPUT DESIGN

Input design in Tkinter involves creating user interfaces that allow users to input data or information into a Python program.

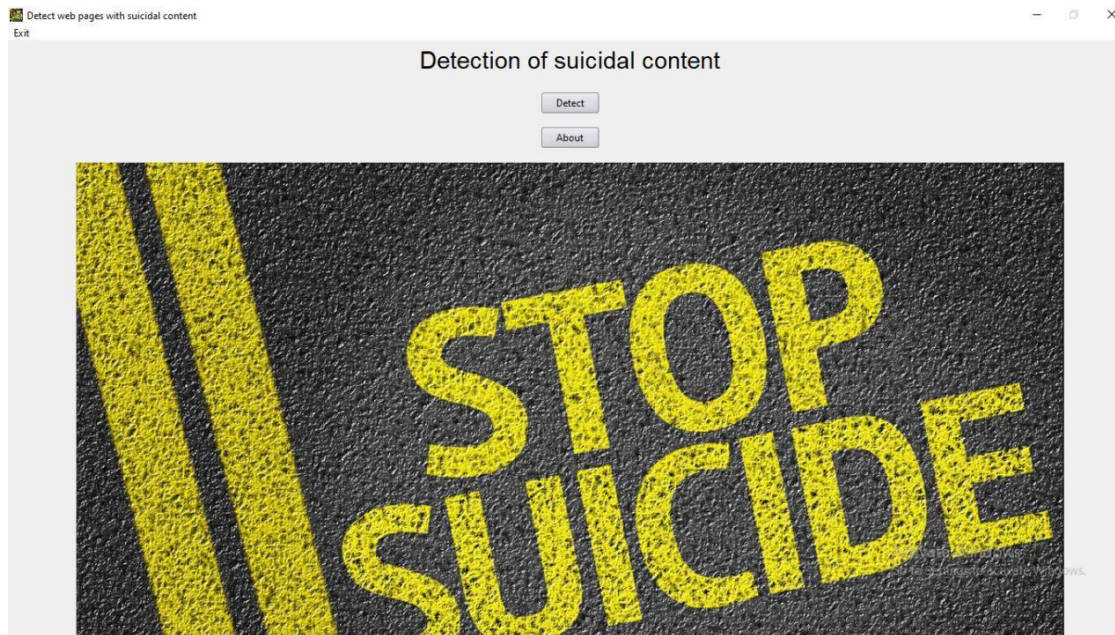


Fig: Input design

This screen indicates input design.

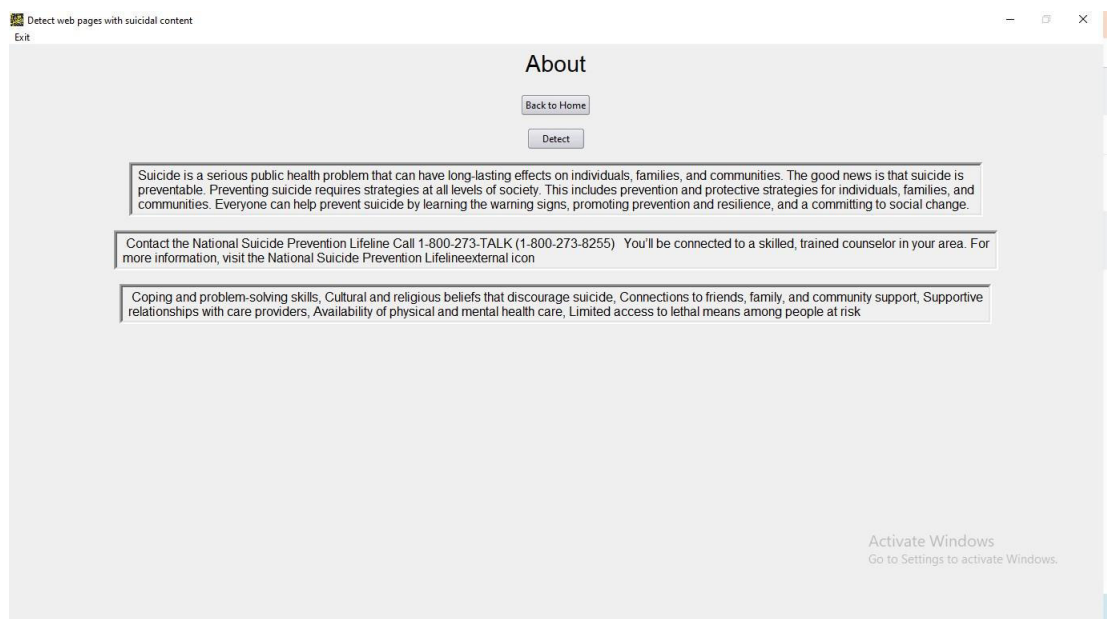


Fig: About screen design

This screen indicates about screen design.



Fig: User Input design

This screen indicates user input design.

OUTPUT SCREEN:

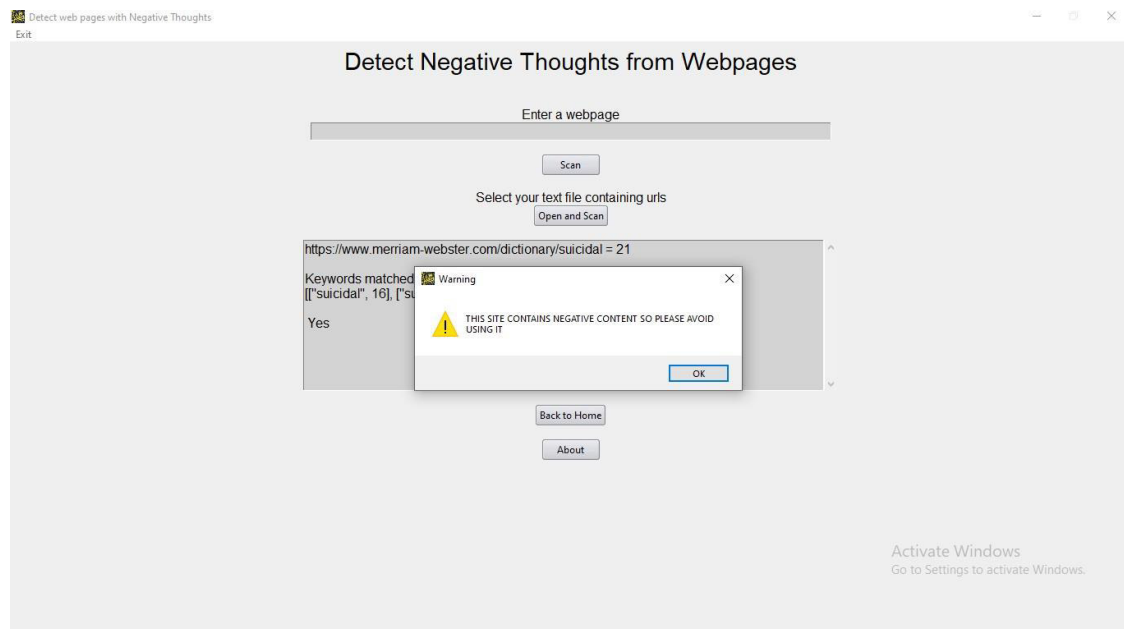


Fig: Output Screen

This screen indicates output screen.

CHAPTER 6

Conclusion and Future Work

In conclusion, the proposed method for detecting suicidal content in social media platforms using natural language processing (NLP) and machine learning techniques offers a promising approach to identifying individuals at risk of suicide at an early stage. By combining keyword-based detection, sentiment analysis, and NLP-based approaches, the method demonstrates effectiveness in recognizing posts that may indicate suicidal ideation. The method's training on a labeled dataset and evaluation using metrics such as precision, recall, and F1-score confirm its accuracy in identifying suicidal content. By utilizing NLP and machine learning techniques, this proposed method addresses the urgent need to identify and intervene with individuals displaying signs of suicidal thoughts. It has the potential to make a significant impact in suicide prevention efforts, enhancing the safety and well-being of individuals within the online community.

FUTURE ENHANCEMENT :

Some potential future enhancements to the system for suicidal content detection using ML in URL could include: Multilingual support: Currently, the system may only be able to detect suicidal content in a specific language. Adding support for multiple languages would allow the system to analyze a wider range of content and improve its accuracy.

Improved feature extraction: Enhancing the system's ability to extract relevant features from webpage content could improve its accuracy in detecting suicidal content. For example, incorporating image analysis or audio analysis could help identify suicidal content that is not only text-based. Real-time analysis: Integrating the system with real-time data sources, such as social media feeds or live chat rooms, could allow for more proactive identification and prevention of suicidal behavior.

Enhanced Natural Language Processing (NLP):

NLP algorithms will become more sophisticated in understanding the nuances of language, including slang, sarcasm, and implicit meanings. This will

enable better detection of suicidal content in text-based platforms such as social media posts, online chat applications.

Multimodal Analysis

Suicidal content detection can be improved by incorporating multimodal analysis, which involves analyzing both text and other modalities like images, audio, and video. Combining textual information with visual and auditory cues can provide a more comprehensive understanding of user intentions and emotions, leading to more accurate detection.

Contextual Analysis

Future models may incorporate contextual analysis techniques to understand the broader context of a conversation or post. By considering factors such as previous interactions, user history, and topic context, machine learning algorithms can better assess the severity and urgency of suicidal content, allowing for more appropriate interventions.

Transfer Learning and Pretrained Models

Transfer learning, where models are pretrained on large-scale datasets and fine-tuned on specific tasks, can significantly enhance the performance of suicidal content detection models. Future work may involve creating specialized models pretrained on extensive collections of labelled suicidal content to provide a strong foundation for subsequent task-specific training.

User-Specific Detection

Machine learning algorithms could be personalized to individual users based on their online behavior and historical data. By learning from an individual's patterns and preferences, the system can better understand their unique communication style and detect potential signs of distress or suicidal ideation that may be specific to that user.

REFERENCES:

1. James SL, Abate D, Abate KH, Abay SM, Abbafati C, Abbasi N, et al. Global, regional, and national incidence, prevalence, and years lived with disability for 354 diseases and injuries for 195 countries and territories, 1990–2017: a systematic analysis for the Global Burden of Disease Study 2017. *Lancet*. (2018) 392:1789–858. doi: 10.1016/S0140-6736(18)32279-7.
2. Kessler RC, Birnbaum H, Bromet E, Hwang I, Sampson N, Shahly V. Age differences in major depression: results from the National Comorbidity Survey Replication (NCS-R). *Psychol Med*. (2010) 40:225–37. doi: 10.1017/S0033291709990213.
3. Hodgetts S, Gallagher P, Stow D, Ferrier IN, O'Brien JT. The impact and measurement of social dysfunction in late-life depression: an evaluation of current methods with a focus on wearable technology. *Int J Geriatr Psychiatry*. (2017) 32:247–55. doi: 10.1002/gps.4632.
4. Fiske A, Wetherell JL, Gatz M. Depression in older adults. *Annu Rev Clin Psychol*. (2009) 5:363–89. doi: 10.1146/annurev.clinpsy.032408.153621.
5. Rodda J, Walker Z, Carter J. Depression in older adults. *BMJ*. (2011) 343:d5219–d5219. doi: 10.1136/bmj.d5219.
6. “Suicide.” <https://www.who.int/news-room/fact-sheets/detail/suicide> (accessed Nov. 24, 2020).
7. “GHO | World Health Statistics data visualizations dashboard | Suicide,” WHO.
8. “Facebook artificial intelligence spots suicidal users - BBC News.” <https://www.bbc.com/news/technology-39126027> (accessed Nov. 24, 2020).

9. “«Рекомендации по распространению в СМИ информации о случаях самоубийства».”
10. [https://www.rospotrebnadzor.ru/documents/details.php?ELEMENT_ID= 6735](https://www.rospotrebnadzor.ru/documents/details.php?ELEMENT_ID=6735) (accessed Nov. 24, 2020).
11. “О деятельности Роспотребнадзора по предостережению самоубийств среди детей и подростков - RSS - Официальный сайт Роспотребнадзора.” http://11.rospotrebnadzor.ru/rss_all/-/asset_publisher/Kq6J/content/id/382348 (accessed Nov. 24, 2020).
12. “Число детских самоубийств в России в 2016 году выросло почти на 60%.” <https://www.interfax.ru/russia/554375> (accessed Nov. 24, 2020).
13. V. Chandler, “Google and suicides: what can we learn about the use of internet to prevent suicides?,” *Public Health*, vol. 154, pp. 144–150, Jan. 2018, doi: 10.1016/j.puhe.2017.10.016.
14. L. Biddle et al., “Suicide and the Internet: Changes in the accessibility of suicide-related information between 2007 and 2014,” *J. Affect. Disord.*, vol. 190, pp. 370–375, Jan. 2016, doi: 10.1016/j.jad.2015.10.028.
15. “Tor Project | Anonymity Online.” <https://www.torproject.org/> (accessed Nov. 27, 2020).

APPENDIX

Source Code:

```
import json # to print list/dict in textbox
import tkinter as tk # root GUI module
import tkinter.scrolledtext as scrolledtext # module for scrollable text widget
import tkinter.ttk as ttk # themed GUI module
from tkinter.filedialog import askopenfile # module to read file
from tkinter import messagebox
import requests # module to get all contents of a website
from bs4 import BeautifulSoup # module to get only text from a website
from PIL import Image, ImageTk # module to open and load a image
from ttkthemes import ThemedStyle # module to use in-built GUI themes
```

```
# class to get all frames together
```

```
class MyApp(tk.Tk):
```

```
    def __init__(self, *args, **kwargs, ):
        tk.Tk._init_(self, *args, **kwargs)
        container = tk.Frame(self)
        container.pack(side="top", fill="both", expand=True)

        container.grid_rowconfigure(0, weight=1)
        container.grid_columnconfigure(0, weight=1)

        self.frames = {}

        menu = tk.Menu(container)

        ex = tk.Menu(menu, tearoff=0)
        menu.add_cascade(menu=ex, label="Exit")
        ex.add_command(label="Exit",
                        command=self.destroy)
```

```

tk.Tk.config(self, menu=menu)

for F in (Startpage, PageOne, PageTwo):
    frame = F(container, self)
    self.frames[F] = frame
    frame.grid(row=0, column=0, sticky="nsew")

self.show_frame(Startpage)

def show_frame(self, cont):
    frame = self.frames[cont]
    frame.tkraise()

# Home page
class Startpage(ttk.Frame):

    def __init__(self, parent, controller):
        ttk.Frame.__init__(self, parent)

        label = ttk.Label(self, text="Detection of Negative Thoughts",
font=("Simplifica", 22)) # page heading
        label.pack(pady=5, padx=5)

        ttk.Label(self, text="").pack()

        button1 = ttk.Button(self, text="Detect ",
                             command=lambda: controller.show_frame(PageOne)) # got to
detect page
        button1.pack()

        ttk.Label(self, text="").pack()

```

```

        button2 = ttk.Button(self, text="About",
                               command=lambda: controller.show_frame(PageTwo)) # got to
about page
        button2.pack()

        ttk.Label(self, text="").pack()

        img = ImageTk.PhotoImage(Image.open(r'wallpaper.jpg').resize((1200,
700))) # set the home page image
        img.image = img
        ttk.Label(self, image=img).pack()

# ***** PAGES *****
# Detect page
class PageOne(ttk.Frame):

    def __init__(self, parent, controller):
        ttk.Frame.__init__(self, parent)
        label = ttk.Label(self, text="Detect Negative Thoughts from Webpages",
font=("Simplifica", 22)) # page heading
        label.pack(pady=5, padx=5)

        ttk.Label(self, text="\n").pack()

        ttk.Label(self, text="Enter a webpage", font=(18)).pack()
        text = tk.Entry(self, font=(26), width=70, bg="lightgray") # textbox to enter a
website
        text.pack()

        ttk.Label(self, text="").pack()

# load the file containing fixed keywords
j = []

```

```

f = open(r'keywords.txt')
for line in f:
    j.append(line.strip())
f.close()
d = dict.fromkeys(j, 0)

# code to scan the website given in textbox
def scan():
    count = 0
    url = text.get()
    text.delete(0, "end")
    result = requests.get(url.strip())
    soup = BeautifulSoup(result.content, 'xml')
    for i in soup.get_text().split():
        if (i.lower() in j):
            count += 1
        if i.lower() in d:
            d[i.lower()] += 1
    l3.config(state=tk.NORMAL)
    l3.delete('1.0', "end")
    di = dict(sorted(d.items(), reverse=True, key=lambda item: item[1]))
    lis = [(k, v) for k, v in di.items() if v >= 1]
    if count == 0:

        l3.insert(tk.END,
            url.strip() + " = " + str(count) + "\n\nKeywords matched: \n" +
            json.dumps(lis) + "\n\n No")
        messagebox.showinfo("Warning", "THIS SITE IS SAFE TO USE
        ENJOY BROWSING ")
    else:
        l3.insert(tk.END,
            url.strip() + " = " + str(count) + "\n\nKeywords matched: \n" +
            json.dumps(lis) + "\n\n Yes")

```

```
        messagebox.showwarning("Warning", "THIS SITE CONTAINS  
NEGATIVE CONTENT SO PLEASE AVOID USING IT")
```

```
l3.config(state=tk.DISABLED)
```

```
b2 = ttk.Button(self, text="Scan", command=scan)  
b2.pack()
```

```
ttk.Label(self, text="").pack()
```

```
# code to open and scan the list of websites given in a text file
```

```
def open_n_scan():
```

```
    files = askopenfile(mode='r', filetypes=[("Text File", "*.txt")])
```

```
    l3.config(state=tk.NORMAL)
```

```
    l3.delete('1.0', "end")
```

```
    for url in files:
```

```
        count = 0
```

```
        result = requests.get(url.strip())
```

```
        soup = BeautifulSoup(result.content, 'lxml')
```

```
        for i in soup.get_text().split():
```

```
            if (i.lower() in j):
```

```
                count += 1
```

```
        l3.insert(tk.END, url.strip() + " = " + str(count) + "\n")
```

```
    l3.config(state=tk.DISABLED)
```

```
ttk.Label(self, text="Select your text file containing urls", font=(18)).pack()
```

```
b1 = ttk.Button(self, text="Open and Scan", command=open_n_scan)
```

```
b1.pack()
```

```
ttk.Label(self, text="").pack()
```

```
l3 = scrolledtext.ScrolledText(self, font=(18), height=10, width=70,  
bg="lightgray",
```

```
state=tk.DISABLED) # multiline textbox
```

```

l3.pack()

ttk.Label(self, text="").pack()

button1 = ttk.Button(self, text="Back to Home",
                      command=lambda: controller.show_frame(Startpage)) # go to
home page
button1.pack()

ttk.Label(self, text="").pack()

button2 = ttk.Button(self, text="About",
                      command=lambda: controller.show_frame(PageTwo)) # got to
about page
button2.pack()

# About page
class PageTwo(ttk.Frame):

    def __init__(self, parent, controller):
        ttk.Frame.__init__(self, parent)
        label = ttk.Label(self, text="About", font=("Simplifica", 22)) # page heading
        label.pack(pady=5, padx=5)

        ttk.Label(self, text="").pack()

        button1 = ttk.Button(self, text="Back to Home",
                              command=lambda: controller.show_frame(Startpage)) # got to
home page
        button1.pack()

        ttk.Label(self, text="").pack()

```

```

button2 = ttk.Button(self, text="Detect",
                      command=lambda: controller.show_frame(PageOne)) # got to
detect page
button2.pack()

ttk.Label(self, text="").pack()

tk.Message(self, relief="sunken", bd=4, font=(20), width=1100, text="Suicide
is a serious public health problem that can have long-
lasting effects on individuals, families, and communities. The good news is that
suicide is preventable. Preventing suicide requires strategies at all levels of
society. This includes prevention and protective strategies for individuals, families,
and communities. Everyone can help prevent suicide by learning the warning
signs, promoting prevention and resilience, and a committing to social
change.").pack()
# Info on suicide
ttk.Label(self, text="").pack()

tk.Message(self, relief="sunken", bd=4, font=(20), width=1100,
text=" Contact the National Suicide Prevention Lifeline Call 1-800-
273-TALK (1-800-273-8255) You'll be connected to a skilled, trained counselor
in your area. For more information, visit the National Suicide Prevention
Lifelineexternal icon").pack()
# Info on suicide content
ttk.Label(self, text="").pack()

tk.Message(self, relief="sunken", bd=4, font=(20), width=1100,
text=" Coping and problem-solving skills, Cultural and religious beliefs
that discourage suicide, Connections to friends, family, and community support,
Supportive relationships with care providers, Availability of physical and mental
health care, Limited access to lethal means among people at risk").pack()
# About
ttk.Label(self, text="").pack()

```

```
ttk.Label(self, text="", font=(20)).pack() # copyright

app = MyApp()

# set default app theme
style = ThemedStyle(app)
style.set_theme("plastik")

# set app icon
icon = ImageTk.PhotoImage(Image.open(r'icon.jpg'))
app.iconphoto(True, icon)

app.resizable(0, 0)
app.title("Detect web pages with Negative Thoughts") # app title
app.state('zoomed') # maximized app by default
app.mainloop()
```

Identification Of Dangerous Posts Based On The Analysis Of Suicidal Content

Abstract—This, according to several suicide studies, there are about 800000 suicides per year and it is still difficult to identify suicidal individuals. Users have openly discussed their attempts to commit suicide and plans on social media platforms as usage of these platforms has increased. Social networks can be used as a source of suicide prevention by identifying profiles that suggest suicidal behavior. First, we analyse social media profiles to extract various elements, such as account features connected to the profile and features related to social media data. An algorithm based on machine learning is used to identify suicidal profiles within Twitter data. After that, we use a data set of suicide victims to build a profile database. Experimental findings attest to the recall and accuracy efficacy of our technique in identifying suicidal characteristics. Finally, we demonstrate our research using a Python-based prototype that demonstrates the identification of suicidal characteristics.

Keywords— *suicide studies, Social media, Python-based prototype, machine learning algorithms, data collection, analytic technique.*

I. INTRODUCTION

According to the World Health Organisation, more than 800,000 individuals take their own lives annually, highlighting the urgent need for suicide prevention efforts. According to the same statistics, Russia is among the top five countries in terms of the suicide rate per 100,000 people. Since the Internet provides the most convenient means of sharing information about suicide on a range of websites pertaining to suicide, many organizations are attempting to discover answers to this problem [1]. Consider the well-known social media platform Facebook, which has the ability to recognize postings with suicidal content and suspect profiles of people who may be suicidal threats.

Federal agencies are working to find a solution. For instance, in Russia, Roskomnadzor developed guidelines in 2016 for media dissemination of information on suicide cases, which is likely affecting search engine rankings on this subject. Additionally, a single register that lists websites restricted in the Russian Federation has existed since 2006 is available. Blocking does not occur instantly, though. Some people are able to access risky websites. Manually censoring suicide websites is scarcely a successful countermeasure to the propagation of suicidal content [2]. After several years of these obstructions, there has been a substantial increase in juvenile

suicides in Russia. One of the reasons could be that, Websites with suicide content have the capacity to make their own copies, or mirrors, in addition to the "death groups" in social networks that platform developers are already aggressively combating [3-4]. In this paper, the possibility of detecting such web pages by real-time content analysis using machine learning algorithms is discussed. At the client's end, detection takes place. This approach makes it possible to identify suicidal individuals early enough to identify the harmful websites they frequently visit.

II. METHODOLOGY

The module description describes each of the modules briefly with its workings and the process. Thus, The following is a description of the aforementioned Image Generating modules.

A. Data Collection

We can document past events by collecting data, and we can use data analysis to find recurring trends. From these patterns, you may build predictive models that use machine learning algorithms to look for trends and anticipate what's going to happen next [5]. Predictive models can only be as accurate as the data used to train them, thus it's crucial to get data in an efficient manner if you want your models to do well. The data must be accurate and complete in order to be used for the task at hand (garbage in, garbage out). For instance, a loan default model might eventually profit from petrol prices but not from changes in tiger populations. In this module, we collect the data from kaggle data set archives. This data set contains the information of divorce in previous years.

B. Data Cleaning

An essential first step in any machine learning effort is data cleaning. In this module, data cleaning is the process of eliminating or changing potentially inaccurate, duplicate, incomplete, or incorrectly formatted data in order to make the data ready for analysis [6]. You can investigate your data using a variety of statistical analysis and data visualization approaches while working with tabular data to find potential data cleaning processes.

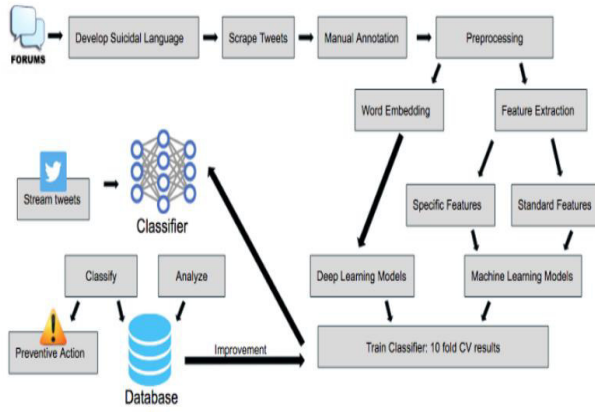


Fig. 1. Architecture and Working of the Process

C. Pre-Processing

In Fig. 1, in order to help with the text's lowercase conversion, punctuation removal, and reduction of related phrases to a single, shared base word, we will build a pre-processing function [7].

D. Exploratory Data Analysis (Eda)

To further comprehend our data, we examined the most frequently used terms from each subeditor using a Count Vectorizer.

1) *Logistic regression*: We should now have a better understanding of machine learning and its variations, I think, thanks to the preceding explanation. One machine learning approach to dealing with classification problems is logistic regression. As a method of predictive analysis, it relies on the concept of probability [8]. If your dependent variable is categorical, you may use the classification technique Logistic Regression to make predictions about it. In logistic regression, the dependent variable is a binary one, meaning that it may take on the values 1 (yes), 0 (no), or 1 (normal, success, etc.). Logistic regression seeks to identify a relationship between a collection of variables and the probability of an event [9]. In Fig. 2, if we want to know how much time a student spent studying for an exam, we may use a two-value response variable: pass or fail. Although logistic regression models are similar to linear ones, they substitute a more complicated cost function called the "sigmoid function" or "logistic function" for the linear one. "Logistic Regression" is the name given to it due to the fact that the method it uses is quite similar to linear regression [10].

The term "Logistic" comes from the Logistic function, which is employed in this categorization method. Why is it not possible to utilize Logistic Regression in place of Linear Regression? In order to better comprehend the notion of linear regression, we shall first explain it from start before responding to this question. Despite its name, logistic regression is a classification approach, even if it is a sister of linear regression [11-13].

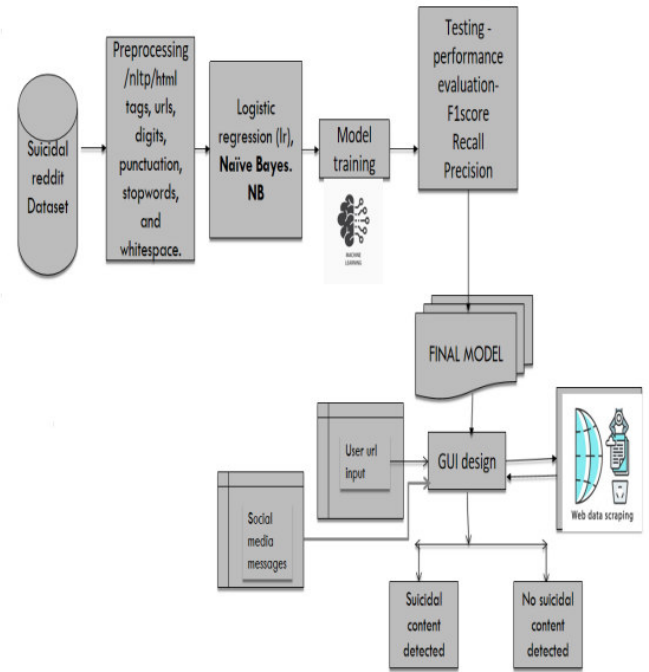


Fig. 2. Working of the Process for Logistic regression

Linear regression in mathematics can be explained by,

$$y = mx + c$$

y is the anticipated value.

m is the line's slope

x is the input data.

C- The slope or Y-intercept

With the help of these values, we can forecast y values. Now, to gain a better idea, look at the diagram below. The blue dots (the input data) stand in for the x values. Our projected line (red line) should now cover most of the places as we can get the slope and y-coordinate using the input data. This line allows us to predict any value of y from its x values. The fact that linear regression is limited to continuous data is something to bear in mind [14]. We will need to make some further adjustments to our algorithm if we wish to incorporate linear regression into our categorization techniques. First, we need to select a threshold, below which our anticipated value falls into class 1 and above which it falls into class 2. Now, if you're thinking, "Hey, that's easy, simply set up a threshold for a linear regression, and voila!, categorized approach," there's a catch. It is not possible to calculate the threshold for large datasets; instead, we have to explicitly define the threshold value [15]. Moreover, the threshold value will not change even if our projected values do. Conversely, A logistic curve with only 0 and 1 as possible values is the result of a logistic regression. Logistic regression makes use of the natural logarithm of the "odds" associated with the target variable, as opposed to linear regression, which creates the curve by chance [16]. Moreover, it is not necessary for the predictors to

have the same variance across all groups or to be normally distributed.

a) *Advantages of Logistic Regression:*

- While over fitting is possible with high-dimensional datasets, it is less common with logistic regression. Regularization (L1 and L2) techniques can be applied in these situations to reduce over-fitting [17].

- It performs well for many simple data sets and has good accuracy when the data set is linearly separable.

- It is easier to train, comprehend, and apply.

- The anticipated parameters, or trained weights, serve as the foundation for the judgements on the significance of each feature. It also indicates if the relationship is positive or negative. Consequently, the association between the characteristics may be discovered using logistic regression [18].

- This method makes it easy to modify models to include new data, unlike decision trees or support vector machines. You can use stochastic gradient descent to update the data.

b) *Disadvantages of Logistic Regression:*

- If there are less observations than features, then logistic regression shouldn't be employed as it could lead to over fitting.

- When working with complex or non-linear data, we won't get better outcomes because it establishes linear bounds.

- It is solely useful for making discrete function predictions. Thus, the discrete number set is the only possible value for the dependent variable in the logistic regression [19].

- Logistic regression requires the average of the independent variables or no multi col-linearity at all.

- To identify every category, logistic regression requires a large data set and a enough number of training samples.

2) *Application of Logistic Regression :*

- Logistic regression covers all use cases where data needs to be divided into numerous groups.

- Credit card fraud detection Spam or ham emails

- Twitter analysis using sentiment analysis

- X-ray and scan image segmentation, recognition, and classification.

- Object detection through video ½. Recognition of handwriting.

- Illness prognosis: diabetes, cancer, Parkinson's disease, etc.

3) *Beautiful Soup:* Web scraping is used to retrieve data from HTML and XML files using a Python tool called Beautiful Soup. It creates a parse tree from the website's source code, from which more readable and hierarchical data

may be retrieved [20]. It is an entire framework made specifically for scraping or web crawling. Additionally, Beautiful Soup, a parsing package, does a respectable job at extracting data from URLs and facilitates the parsing of particular sections of them. Before ending, it merely retrieves the material from the supplied URL. A technique for retrieving data from the web is web data extraction, sometimes called web scraping. The majority of this data is HTML information that is transformed into structured data in a database or spreadsheet before being used in various applications [21].

4) *Tkinter:* A popular Python library for creating graphical user interfaces is called Tkinter. With Tkinter, creating a graphical user interface is incredibly simple and quicker. Several widgets from Tkinter can be used to create graphical user interfaces. These consist of check boxes, radio buttons, buttons, etc. Continuous data processing is necessary for machine learning, and Python's modules enable you to access, handle, and modify data [22]. Some of the most popular libraries available for ML and AI are as follows: For managing fundamental machine learning methods such as clustering, regression, classification, linear and logistic regressions, and others, use Scikit-Learn. For data science, Tkinter provides access to the geometric configuration of the widgets that arrange the widgets in parent windows in Python. There are primarily three kinds for geometry managers.

III. RESULTS AND DISCUSSION

ROC diagram showing the process by which algorithms are taught on a training set. Based on the provided comprehensive metrics of the trained algorithms, we may infer that SGD has the best accuracy while Naive Bayes has the lowest. The data collected from the control group's website algorithm testing. We may infer from the data that although the algorithms do a decent job of identifying malicious websites on the whole, computational errors do sometimes lead them to incorrectly label safe websites as dangerous. Another way of looking at it is that a 50% threshold yields acceptable detection results, whereas a 10% barrier drastically reduces the accuracy of detection.

Thus, machine learning algorithms correctly identify fraudulent websites 46 times out of 50, which is a good result, given a threshold of 50%.

IV. FUTURE WORK

A. *Enhanced Natural Language Processing (NLP)*

NLP algorithms will become more sophisticated in understanding the nuances of language, including slang, sarcasm, and implicit meanings. This will enable better detection of suicidal content in text-based platforms such as social media posts, online chat applications.

B. *Multi-modal Analysis*

Suicidal content detection can be improved by incorporating multi-modal analysis, which involves analyzing both text and other modalities like images, audio, and video.

Combining textual information with visual and auditory cues can provide a more comprehensive understanding of user intentions and emotions, leading to more accurate detection.

C. Contextual Analysis

Future models may incorporate contextual analysis techniques to understand the broader context of a conversation or post. By considering factors such as previous interactions, user history, and topic context, machine learning algorithms can better assess the severity and urgency of suicidal content, allowing for more appropriate interventions.

D. Transfer Learning and Pretrained Models

Transfer learning can greatly improve the performance of suicidal content detection algorithms by pretrained them on large-scale datasets and fine-tuning them on particular tasks.

Future work may involve creating specialized models pretrained on extensive collections of labeled suicidal content to provide a strong foundation for subsequent task-specific training.

1) *User-Specific Detection*: Machine learning algorithms could be personalized to individual users based on their online behavior and historical data. By learning from an individual's patterns and preferences, the system can better understand their unique communication style and detect potential signs of distress or suicidal ideation that may be specific to that user.

2) *Collaborative Efforts*: Collaboration between mental health professionals, researchers, and technology companies could greatly enhance the development of effective suicidal content detection systems. Combining domain expertise with machine learning approaches can lead to more accurate and contextually relevant solutions.

3) *Ethical Considerations*: As technology advances, it becomes crucial to address the ethical implications of suicidal content detection. Striking a balance between user privacy, data security, and the need to intervene in potentially life-threatening situations is a significant challenge that future research should address. Fourthly, expanding the system's coverage to include additional data sources, such as social media and public records, to enhance tracking capabilities. Lastly, ongoing user training and awareness programs to ensure effective utilization of the system by law enforcement and the public, ultimately increasing the chances of reuniting missing persons with their families.

V. CONCLUSION

Machine learning algorithms were used in this study to detect Websites with suicidal content that should be avoided. This article describes This method detects dangerous web pages based on a number of factors experimental results. This method can be successfully used in real life based on the obtained results of detecting the control group. The following improvements will be implemented in the future for the system used to check web pages for suicidal content: It is necessary to verify the images, since some of these images may contain symbols of death groups or be suicidal. In the last

case, the website contains it is likely to be part of the group. Check for instructions on how to commit suicide. Further examination of text processing libraries and machine learning techniques in the future we intend to improve this method. It is worthwhile to continue researching this area so that dangerous web pages can be detected more accurately.

REFERENCES

- [1] James SL, Abate D, Abate KH, Abay SM, Abbafati C, Abbasi N, et al. Global, regional, and national incidence, prevalence, and years lived with disability for 354 diseases and injuries for 195 countries and territories, 1990–2017: a systematic analysis for the Global Burden of Disease Study 2017. *Lancet*. (2018) 392:1789–858. doi: 10.1016/S0140-6736(18)32279-7.
- [2] Kessler RC, Birnbaum H, Bromet E, Hwang I, Sampson N, Shahly V. Age differences in major depression: results from the National Comorbidity Survey Replication (NCS-R). *Psychol Med*. (2010) 40:225–37. doi: 10.1017/S0033291709990213.
- [3] Hodgetts S, Gallagher P, Stow D, Ferrier IN, O'Brien JT. The impact and measurement of social dysfunction in late-life depression: an evaluation of current methods with a focus on wearable technology. *Int J Geriatr Psychiatry*. (2017) 32:247–55. doi: 10.1002/gps.4632.
- [4] Fiske A, Wetherell JL, Gatz M. Depression in older adults. *Annu Rev Clin Psychol*. (2009) 5:363–89. doi: 10.1146/annurev.clinpsy.032408.153621.
- [5] Rodda J, Walker Z, Carter J. Depression in older adults. *BMJ*. (2011) 343:d5219–d5219. doi: 10.1136/bmj.d5219.
- [6] “Suicide.” <https://www.who.int/news-room/fact-sheets/detail/suicide> (accessed Nov. 24, 2020).
- [7] “GHO | World Health Statistics data visualizations dashboard | Suicide,” WHO.
- [8] “Facebook artificial intelligence spots suicidal users - BBC News.” <https://www.bbc.com/news/technology-39126027> (accessed Nov. 24, 2020).
- [9] “Rekomendacii po rasprostraneniyu v SMI informacii o sluchayah samoubijstva”.
- [10] https://www.rosпотреbnadzor.ru/documents/details.php?ELEMENT_ID=6735 (accessed Nov. 24, 2020).
- [11] “O deyatel'nosti Rospotrebnadzora po predotvrashcheniyu samoubijstv sredi detej i podrostkov - RSS - Oficial'nyj sayt Rospotrebnadzora.” http://11.rosпотреbnadzor.ru/rss_all/asset_publisher/Kq6J/content/id/382348 (accessed Nov. 24, 2020).
- [12] “Chislo detskikh suicidov v Rossii v 2016 godu vyroslo pochni na 60%.” <https://www.interfax.ru/russia/554375> (accessed Nov. 24, 2020).
- [13] V. Chandler, “Google and suicides: what can we learn about the use of internet to prevent suicides?,” *Public Health*, vol. 154, pp. 144–150, Jan. 2018, doi: 10.1016/j.puhe.2017.10.016.
- [14] L. Biddle et al., “Suicide and the Internet: Changes in the accessibility of suicide-related information between 2007 and 2014,” *J. Affect. Disord.*, vol. 190, pp. 370–375, Jan. 2016, doi: 10.1016/j.jad.2015.10.028.
- [15] “Tor Project | Anonymity Online.” <https://www.torproject.org/> (accessed Nov. 27, 2020).
- [16] C. M. Mörch, L. P. Côté, L. Corthésy-Blondin, L. Plourde-Léveillé, L. Dargis, and B. L. Mishara, “The Darknet and suicide,” *J. Affect. Disord.*, vol. 241, pp. 127–132, Dec. 2018, doi: 10.1016/j.jad.2018.08.028.
- [17] K. P. Kumar, N. Jaisankar, and N. Mythili, “An efficient technique for detection of suspicious malicious web site,” *Journal of Advances in Information Technology*, vol. 2, no. 4, 2011.
- [18] B. Feinstein, D. Peck, and I. Secure Works, “Caffeine monkey: Automated collection, detection and analysis of malicious javascript,” *Black Hat USA*, vol. 2007, 2007.
- [19] C. N. Dos Santos and M. Gatti, “Deep convolutional neural networks for sentiment analysis of short texts,” in *COLING*, pp. 69–78, 2014.
- [20] X. Zhang, J. Zhao, and Y. LeCun, “Character-level convolutional networks for text classification,” in *Advances in neural information processing systems*, pp. 649–657, 2015.

- [21] Vioulès, Mia & Moulahi, Bilel & Azé, Jérôme & Bringay, Sandra. (2017). Detection of Suicide-Related Posts in Twitter Data Streams. IBM Journal of Research and Development. 62. 10.1147/JRD.2017.2768678.
- [22] A. Schoene, L. Bojanic, M. Nghiem, I. Hunt and S. Ananiadou, "Classifying Suicide-Related Content and Emotions on Twitter Using Graph Convolutional Neural Networks" in IEEE Transactions on Affective Computing, vol. 14, no. 03, pp. 1791-1802, 2023. doi: 10.1109/TAFFC.2022.3221683.