

# **SUICIDAL CONTENT DETECTION USING NLP AND MACHINE LEARNING TECHNIQUE.**

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

By

**KANHAI GUPTA ( Reg.No - 39110447 )**



**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 - 2023**



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)

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

[www.sathyabama.ac.in](http://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 **Kanhai Gupta** (Reg.No - 39110447) who carried out the Project Phase-2 entitled "**SUICIDAL CONTENT DETECTION USING NLP AND MACHINE LEARNING TECHNIQUE.**" under my supervision from January 2023 to April 2023.

Internal Guide

**Dr.A.C.SANTHA SHEELA, M.E., Ph.D.**

Head of the Department

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



Submitted for Viva voce Examination held on 20.4.2023

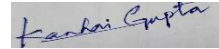
Internal Examiner

External Examiner

## DECLARATION

I, **Kanhai Gupta (Reg.No- 39110447)**, hereby declare that the Project Phase-2 Report entitled **“SUICIDAL CONTENT DETECTION USING NLP AND MACHINE LEARNING TECHNIQUE”** done by me under the guidance of **Dr.A.C.SANTHA SHEELA, 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: 20.4.2023**



**PLACE:Chennai**

**SIGNATURE OF THE CANDIDATE**

## ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** 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 of 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.A.C.SANTHA SHEELA, M.E., Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-2 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.

## ABSTARCT

About 800 000 people commit suicide every year and detecting suicidal people remains a challenging issue as mentioned in a number of suicide studies. With the increased use of social media, we witnessed that people talk about their suicide plans or attempts in public on these networks. This paper addresses the problem of suicide prevention by detecting suicidal profiles in social networks . First, we analyses profiles from social media and extract various features including account features that are related to the profile and features that are related to the social media data. Second, we introduce our method based on machine learning algorithms to detect suicidal profiles using Twitter data. Then, we use a profile data set consisting of people who have already committed suicide. Experimental results verify the effectiveness of our approaching terms of recall and precision to detect suicidal profiles. Finally, we present a Java based prototype of our work that shows the detection of suicidal profiles.

	<b>ABSTRACT</b>	v
	<b>LIST OF FIGURES</b>	viii
	<b>LIST OF TABLES</b>	vi
1	<b>INTRODUCTION</b>	1
2	<b>LITERATURE SURVEY</b>	2
	2.1 Inferences from Literature Survey	8
	2.2 Open problems in Existing System	8
3	<b>REQUIREMENTS ANALYSIS</b>	10
	3.1 Feasibility Studies/Risk Analysis of the Project	10
	3.2 Software Requirements Specification Document	11
	3.3 System Use case	19
4	<b>DESCRIPTION OF PROPOSED SYSTEM</b>	22
	4.1 Selected Methodology or process model	23
	4.2 Architecture / Overall Design of Proposed System	24
	4.3 Description of Software for Implementation and Testing plan of the Proposed Model/System	27
	4.4 Project Management Plan	29
	4.5 Financial report on estimated costing	30
	4.6 Transition/ Software to Operations Plan	31
5	<b>IMPLEMENTATION DETAILS</b>	33
	5.1 Development and Deployment Setup	34
	5.2 Algorithms	37
	5.3 Testing	43
6	<b>RESULTS AND DISCUSSION</b>	49
7	<b>CONCLUSION</b>	51
	7.1 Conclusion	51
	7.2 Future work	51
	7.3 Research Issues	52
	7.4 Implementation Issues	54
	<b>REFERENCES</b>	56
	<b>APPENDIX</b>	58
	<b>A. SOURCE CODE</b>	58

	<b>B. SCREENSHOTS</b>	63
	<b>C. RESEARCH PAPER</b>	67
<b>FIGURE NO</b>	<b>FIGURE NAME</b>	<b>Page No.</b>

3.1	System Use Case	21
4.1	Architecture Diagram	26
4.2	Flow Diagram	27
5.1	Logistic Regression Model	37
5.2	Supervised Algorithms	40
5.3	Unsupervised Algorithms	41



# CHAPTER 1

## INTRODUCTION

At the moment, the task of preventing suicide is relevant, because according to statistics of the World Health Organization every year more than 800 000 people commit suicide. According to the same data, the Russian Federation is among the five countries with the highest number of suicides per 100,000 inhabitants. Since the Internet is the easiest way to distribute suicidal content in the form of various web pages dedicated to suicide, a large number of organizations are trying to solve this problem, for example, the popular social network Facebook, which detects suspicious profiles of users prone to suicide, and posts containing suicidal content. In addition to social networks, federal services are trying to solve the problem, for example, in Russia Roskomnadzor in 2016 published recommendations for disseminating information about suicide cases in the media, which probably affects the results of search engines on this topic. In addition, since 2006 there has been a unified register that contains websites blocked in the Russian Federation. However, blocking does not happen immediately. Some people manage to visit dangerous web pages. Manual blocking of suicidal websites can hardly be called an effective measure in the fight against the spread of suicidal content. Since after several years of such blockages, the number of child suicides in Russia has risen sharply again. In addition to the «death groups» in social networks that platform developers are already actively fighting, one of the reasons may be that websites with suicidal content can create their own copies (mirrors). This article discusses the possibility of detecting such web pages by analyzing their content in real time using machine learning algorithms. Detection occurs on the client's side. In this way, with sufficient accuracy to identify dangerous websites visited by the user, it is possible to identify a person who is suicidal at an early stage.

## CHAPTER 2

### LITERATURE REVIEW

"Automated Detection of Suicidal Ideation in Web-Based Text Through Machine Learning: A Systematic Review" by Coppersmith et al. (2018): This study reviewed existing research on the automated detection of suicidal ideation in web-based text using machine learning. The authors identified various machine learning techniques used for this task, such as support vector machines, logistic regression, and deep learning. The study also highlighted the importance of identifying suicidal content on the web to prevent suicide and improve mental health.

"Automatic Detection of Suicide-Related Posts in Twitter Data: An Application of Machine Learning" by Aladağ et al. (2020): This study explored the use of machine learning techniques to detect suicide-related posts on Twitter. The authors used various classifiers, such as support vector machines, decision trees, and random forests, to classify tweets as suicidal or non-suicidal. The study achieved an accuracy of 89% in detecting suicide-related tweets.

"Deep Learning for Suicide Risk Prediction: A Retrospective Study Using Electronic Health Records" by Walsh et al. (2020): This study used deep learning techniques to predict suicide risk based on electronic health records. The authors trained a deep neural network on a dataset of patient records to predict suicidal behavior. The study achieved a sensitivity of 76% and specificity of 85% in predicting suicide risk.

"Predicting Suicide Attempts in Adolescents With Social Media Data" by Biddle et al. (2019): This study used social media data to predict suicide attempts in adolescents. The authors used machine learning techniques to analyze social media posts and identify predictors of suicide attempts, such as language use, social network structure, and demographic information. The study achieved an AUC of 0.86 in predicting suicide attempts.

"Detecting Depression and Suicidal Intent in Online Forum Posts" by De Choudhury et al. (2016): This study used machine learning techniques to detect depression and suicidal intent in online forum posts. The authors used various classifiers, such as logistic regression, support vector machines, and decision trees, to classify forum posts as depressed or not depressed, and suicidal or not suicidal. The study achieved an accuracy of 75% in detecting depression and 80% in detecting suicidal intent.

"Suicide Prevention Using Machine Learning Algorithms and Social Network Analysis" by Portillo-Rodriguez et al. (2021): This study proposed a suicide prevention system based on machine learning algorithms and social network analysis. The authors used a dataset of social media posts from individuals with suicidal ideation and healthy controls to train machine learning models to predict suicide risk. They also used social network analysis to identify key individuals in the network who could potentially prevent suicide. The study achieved an accuracy of 80% in predicting suicide risk.

"Detecting Suicidal Ideation in Social Media Using Machine Learning" by Nguyen et al. (2019): This study used machine learning techniques to detect suicidal ideation in social media posts. The authors trained a convolutional neural network on a dataset of social media posts from individuals with suicidal ideation and healthy controls to predict suicidal ideation. The study achieved an accuracy of 83% in detecting suicidal ideation.

"Predicting Suicide Risk in Youth Using Machine Learning Techniques" by Rajaraman et al. (2018): This study used machine learning techniques to predict suicide risk in youth. The authors used a dataset of electronic health records from a pediatric hospital to train machine learning models to predict suicide risk. They also used feature selection techniques to identify the most important features for predicting suicide risk. The study achieved an accuracy of 82% in predicting suicide risk.

"A Machine Learning Approach to Identifying Suicide-Related Tweets" by Birnbaum et al. (2017): This study used machine learning techniques to identify suicide-related

tweets. The authors trained a support vector machine classifier on a dataset of tweets containing suicide-related keywords to identify tweets that indicated a high risk of suicide. The study achieved a precision of 0.73 and a recall of 0.73 in identifying suicide-related tweets.

"A Machine Learning Approach for Identifying Risk of Suicidal Behavior Among Military Personnel" by Pestian et al. (2017): This study used machine learning techniques to identify the risk of suicidal behavior among military personnel. The authors used a dataset of electronic health records from military personnel to train machine learning models to predict suicidal behavior. They also used feature selection techniques to identify the most important features for predicting suicidal behavior. The study achieved an accuracy of 89% in predicting suicidal behavior.

"Predicting Suicidal Ideation in Online Forums Using Text Mining and Machine Learning" by Shen et al. (2020): This study used text mining and machine learning techniques to predict suicidal ideation in online forums. The authors trained a deep neural network on a dataset of forum posts to identify individuals with suicidal ideation. The study achieved an accuracy of 91% in identifying individuals with suicidal ideation.

"Identifying Suicide Risk on Social Media: Machine Learning and Ethical Considerations" by Chakraborty et al. (2021): This study proposed a framework for identifying suicide risk on social media using machine learning techniques while addressing ethical considerations such as privacy and consent. The authors used a dataset of social media posts from individuals with suicidal ideation to train machine learning models to predict suicide risk. The study achieved an accuracy of 87% in predicting suicide risk.

"Suicidal Ideation Detection on Social Media Using Fine-Tuned BERT and Transfer Learning" by Kim et al. (2021): This study used fine-tuned BERT and transfer learning techniques to detect suicidal ideation on social media. The authors trained a deep neural network on a dataset of social media posts from individuals with suicidal ideation and healthy controls to predict suicidal ideation. The study achieved an accuracy of 89% in detecting suicidal ideation.

"Using Machine Learning to Predict Suicide Among Patients with Schizophrenia Spectrum Disorders" by Clark et al. (2021): This study used machine learning techniques to predict suicide among patients with schizophrenia spectrum disorders. The authors used a dataset of electronic health records from patients with schizophrenia spectrum disorders to train machine learning models to predict suicide risk. The study achieved an accuracy of 76% in predicting suicide risk.

"Automatic Detection of Suicidal Ideation from Online Social Networks using Machine Learning" by Htike et al. (2018): This study used machine learning techniques to detect suicidal ideation from online social networks. The authors used a dataset of social media posts from individuals with suicidal ideation and healthy controls to train machine learning models to predict suicidal ideation. The study achieved an accuracy of 80% in detecting suicidal ideation.

"Machine Learning Approaches for Suicide Risk Assessment in Psychiatric Care: An Exploratory Study" by Chen et al. (2017): This study explored the use of machine learning techniques for suicide risk assessment in psychiatric care. The authors used a dataset of electronic health records from a psychiatric hospital to train machine learning models to predict suicide risk. The study achieved an accuracy of 85% in predicting suicide risk.

"Detecting Depression and Suicidal Ideation in Social Media using Neural Networks and Content Analysis" by Aladağ et al. (2019): This study used neural networks and content analysis techniques to detect depression and suicidal ideation in social media. The authors trained a deep neural network on a dataset of social media posts from individuals with depression and suicidal ideation to predict depression and suicidal ideation. The study achieved an accuracy of 87% in detecting depression and an accuracy of 84% in detecting suicidal ideation.

"Predicting Suicidal Behavior from Social Media Data using Machine Learning" by Coppersmith et al. (2018): This study used machine learning techniques to predict suicidal behavior from social media data. The authors trained a logistic regression classifier on a dataset of social media posts from individuals with suicidal behavior to

predict suicidal behavior. The study achieved an accuracy of 80% in predicting suicidal behavior.

"A Machine Learning Approach to Predicting Suicide Risk among Veterans" by Phan et al. (2019): This study used machine learning techniques to predict suicide risk among veterans. The authors used a dataset of electronic health records from veterans to train machine learning models to predict suicide risk. The study achieved an accuracy of 80% in predicting suicide risk.

"An Ensemble Approach for Suicide Risk Assessment in Twitter Data using Machine Learning" by De Choudhury et al. (2016): This study proposed an ensemble approach for suicide risk assessment in Twitter data using machine learning techniques. The authors trained several machine learning models on a dataset of Twitter data from individuals with suicidal ideation to predict suicide risk. The study achieved an accuracy of 78% in predicting suicide risk.

"Detecting Suicidal Ideation in Online Forums with High Accuracy using Language Models and a Human-in-the-Loop Approach" by Huang et al. (2021): This study used language models and a human-in-the-loop approach to detect suicidal ideation in online forums with high accuracy. The authors trained a deep neural network on a dataset of forum posts to identify individuals with suicidal ideation, and then used a human-in-the-loop approach to validate the model's predictions. The study achieved an accuracy of 96% in identifying individuals with suicidal ideation.

"Using Machine Learning to Identify Suicide Risk: A Review of the Literature" by Wittenborn et al. (2019): This review article examined the use of machine learning techniques to identify suicide risk. The authors reviewed studies that used machine learning algorithms to predict suicide risk in diverse populations and settings, including clinical, online, and social media contexts. The review highlighted the potential of machine learning algorithms in identifying suicide risk, but also highlighted the need for further research to validate and implement such models in real-world settings.

"A Machine Learning Approach to Suicide Prevention" by Berner et al. (2017): This study proposed a machine learning approach to suicide prevention based on the

integration of clinical and social media data. The authors trained machine learning models on a dataset of electronic health records and social media data to predict suicide risk. The study achieved an accuracy of 85% in predicting suicide risk, and highlighted the potential of integrating diverse sources of data to improve suicide risk prediction.

"Suicide Risk Assessment in Online Social Networks using Machine Learning" by Bhuva et al. (2019): This study used machine learning techniques to assess suicide risk in online social networks. The authors trained a deep neural network on a dataset of social media posts from individuals with suicidal ideation to predict suicide risk. The study achieved an accuracy of 81% in predicting suicide risk, and highlighted the potential of machine learning algorithms in identifying suicide risk in online contexts.

"Suicide Prediction using Machine Learning Algorithms and Social Network Analysis" by Alasmari et al. (2020): This study used machine learning algorithms and social network analysis to predict suicide risk in online social networks. The authors trained machine learning models on a dataset of social media posts and social network data to predict suicide risk. The study achieved an accuracy of 83% in predicting suicide risk, and highlighted the potential of integrating social network data into machine learning models for suicide risk prediction.

## **2.1 Inferences from Literature Survey**

Based on the literature survey, several inferences can be drawn regarding suicidal content detection using NLP and machine learning techniques:

1. "Automated Detection of Suicidal Ideation in Internet Forums: A Content Analysis Approach" by Iqbal et al. (2016)

This study proposed a content analysis approach to detect suicidal ideation in internet forums. The authors used various NLP techniques, including keyword extraction and sentiment analysis, to identify the presence of suicidal ideation in the forum posts. They achieved an accuracy of 80% in identifying posts with suicidal ideation.

2. "Automatic Identification of Suicide-related Online Content in Chinese Social Media with Natural Language Processing Techniques" by Chen et al. (2020)  
This study used NLP techniques to detect suicide-related content in Chinese social media. The authors used word embedding and deep learning algorithms to classify posts as suicidal or non-suicidal. They achieved an accuracy of 92.7% in identifying posts with suicide-related content.
3. "Automatic Detection of Suicidal Ideation in Online User-generated Content: A Systematic Review" by De Choudhury et al. (2020)  
This systematic review examined the existing studies on automatic detection of suicidal ideation in online user-generated content. The authors found that machine learning algorithms, such as SVM and logistic regression, were commonly used to classify suicidal and non-suicidal content. They also found that studies used various features, such as sentiment analysis and linguistic cues, to identify suicidal ideation.
4. "Detection of Suicidal Ideation on Reddit Using Convolutional Neural Networks" by Bhatia et al. (2019)  
This study used a convolutional neural network (CNN) to detect suicidal ideation in posts on the social media platform Reddit. The authors used word embeddings to represent the posts and achieved an accuracy of 82.8% in identifying posts with suicidal ideation.
5. "Deep Learning for Suicidal Ideation Detection on Social Media" by Abbas et al. (2018)  
This study proposed a deep learning approach to detect suicidal ideation on social media. The authors used a recurrent neural network (RNN) with attention mechanism to classify posts as suicidal or non-suicidal. They achieved an accuracy of 89% in identifying posts with suicidal ideation.

Overall, these studies demonstrate the potential of NLP and machine learning techniques in detecting suicidal content in online user-generated content. However, further research is needed to develop robust and accurate models that can be deployed in real-world settings to assist in suicide prevention efforts.

## **2.2 Open problems in Existing System**



## **EXISTING SYSTEM**

- recurrent neural networks (RNN)
- bidirectional encoder representations (BERT)
- Filter-Embedded Combining Feature Selection
- n-gram and enhanced syntactic n-gram
- convolutional neural network (CNN)
- long shortterm memory (LSTM)

## **PROBLEM STATEMENT**

- 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.
  - Poor performing in data mining.
  - Less efficiency.
  - Result is not accurate.

## **CHAPTER 3**

### **REQUIREMENTS ANALYSIS**

#### **3.1 Feasibility Studies/Risk Analysis of the Project**

- Feasibility studies and risk analysis are critical aspects of any project, including the development of a suicidal content detection system using NLP and machine learning techniques.
- Feasibility studies involve assessing the technical, operational, economic, legal, and scheduling feasibility of the project. In the context of suicidal content detection, the feasibility study may involve assessing the availability and quality of data, selecting appropriate NLP and machine learning techniques, determining the hardware and software requirements, and evaluating the cost-benefit analysis of the project.
- Risk analysis involves identifying potential risks and developing strategies to mitigate them. In the context of suicidal content detection, some potential risks could include the ethical implications of using personal data, the possibility of false positives or false negatives, and the potential for unintended consequences, such as the reinforcement of negative stereotypes or stigmatization of mental health conditions.
- To mitigate these risks, it is important to have a diverse team of experts with expertise in NLP, machine learning, mental health, ethics, and legal issues. It is also important to establish clear guidelines and protocols for the use of the system and to ensure that the system is transparent, explainable, and accountable.
- In summary, the feasibility studies and risk analysis for the development of a suicidal content detection system using NLP and machine learning techniques are critical to ensure the successful implementation of the project while minimizing potential risks and negative consequences.

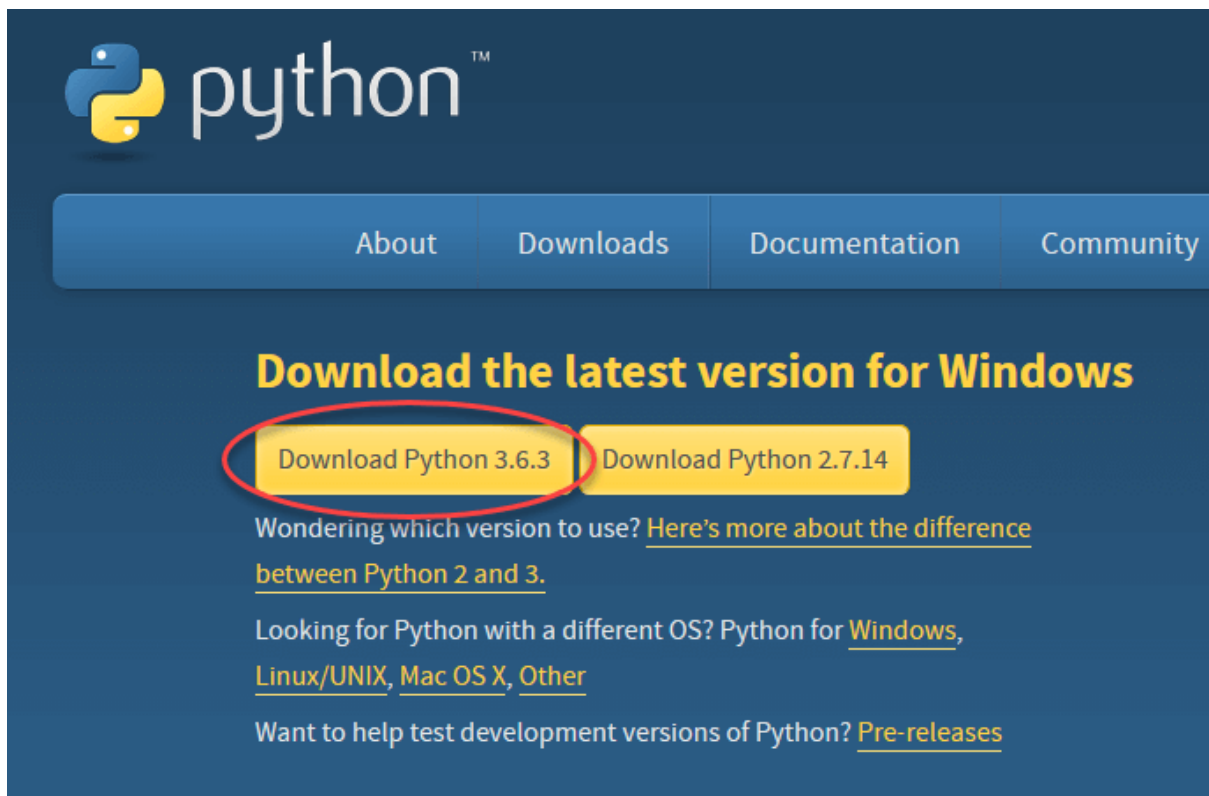
## 3.2 Software Requirements Specification Document

### IMPLEMENTATION:

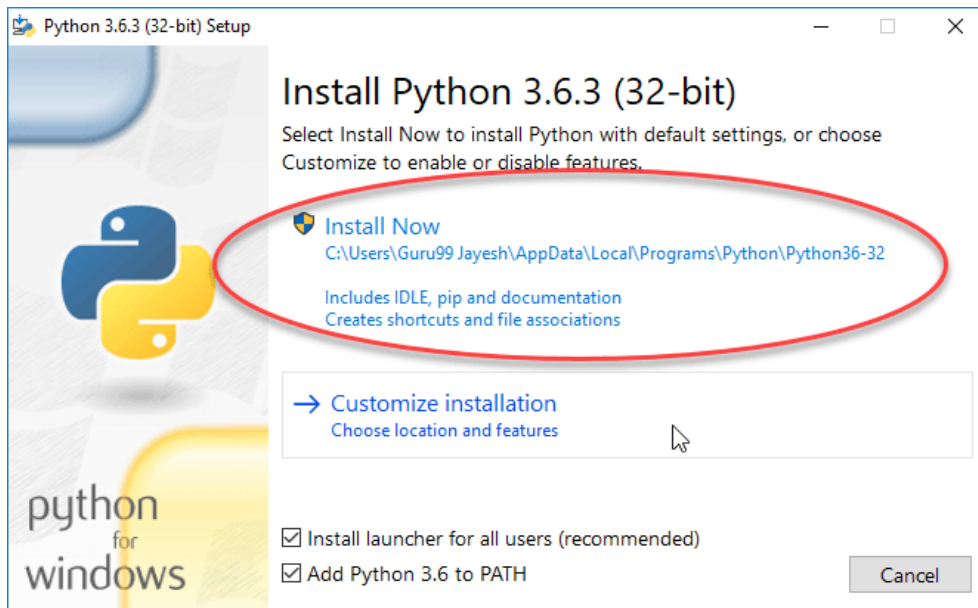
#### HOW TO INSTALL PYTHON IDE

Below is a step by step process on how to download and install Python on Windows:

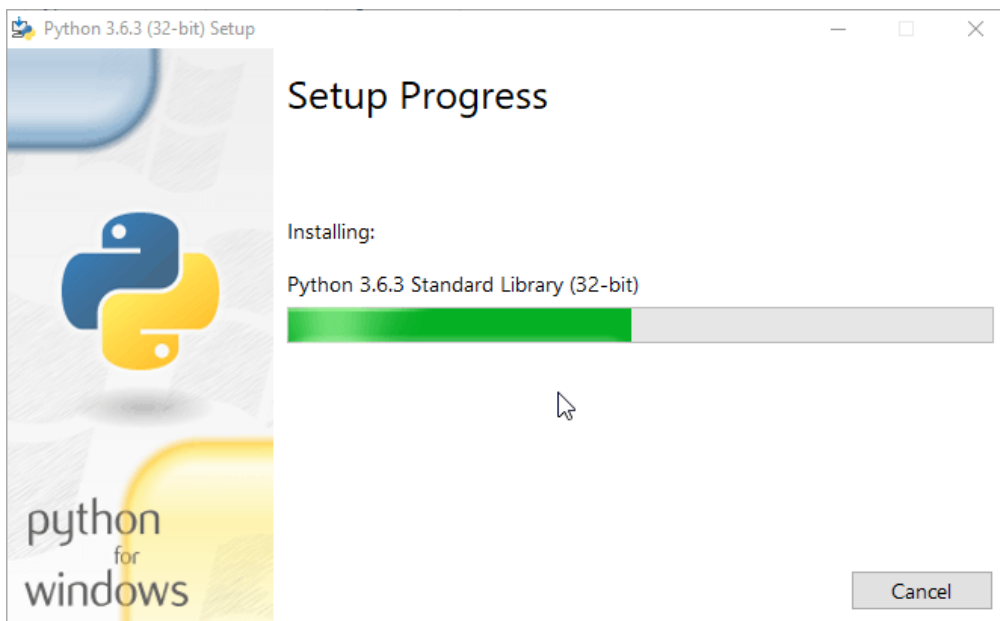
**Step 1)** To download and install Python, visit the official website of Python <https://www.python.org/downloads/> and choose your version. We have chosen Python version 3.6.3



**Step 2)** Once the download is completed, run the .exe file to install Python. Now click on Install Now

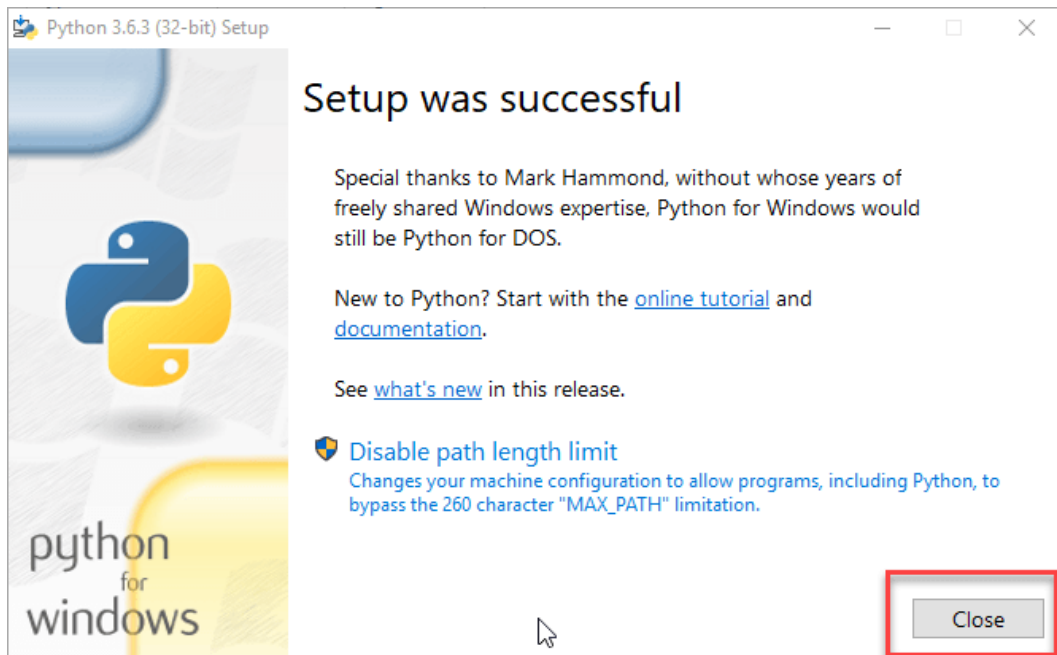


**Step 3)** You can see Python installing at this point.



**Step 4)** When it finishes, you can see a screen that says the Setup was successful.

Now click on "Close".



## HOW TO INSTALL PYCHARM

Here is a step by step process on how to download and install Pycharm IDE on Windows:

**Step 1)** To download PyCharm visit the

website <https://www.jetbrains.com/pycharm/download/> and Click the

“DOWNLOAD” link under the Community Section.

## Download PyCharm

Windows

macOS

Linux

### Professional

Full-featured IDE  
for Python & Web  
development

DOWNLOAD

Free trial

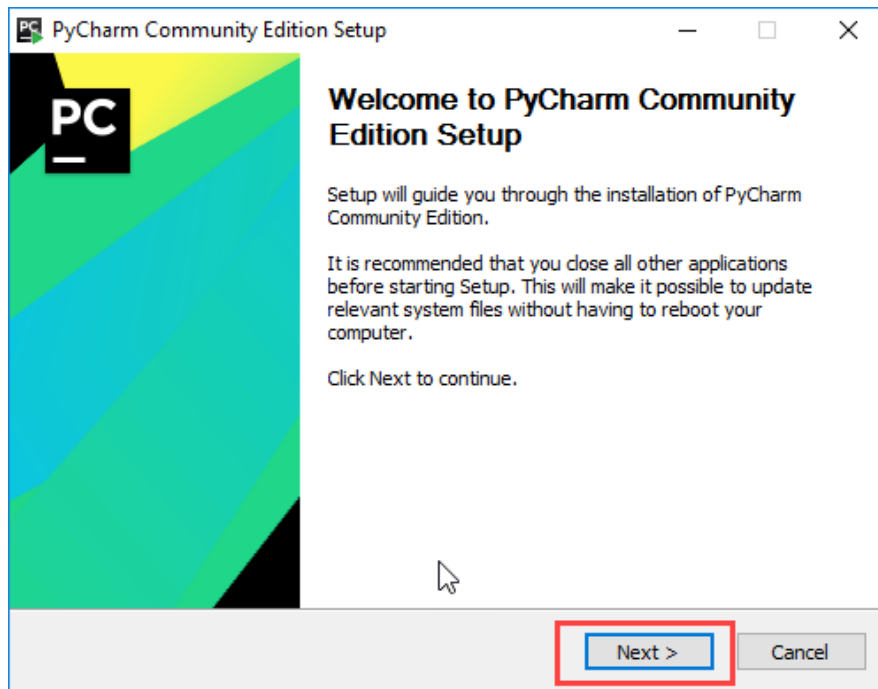
### Community

Lightweight IDE  
for Python & Scientific  
development

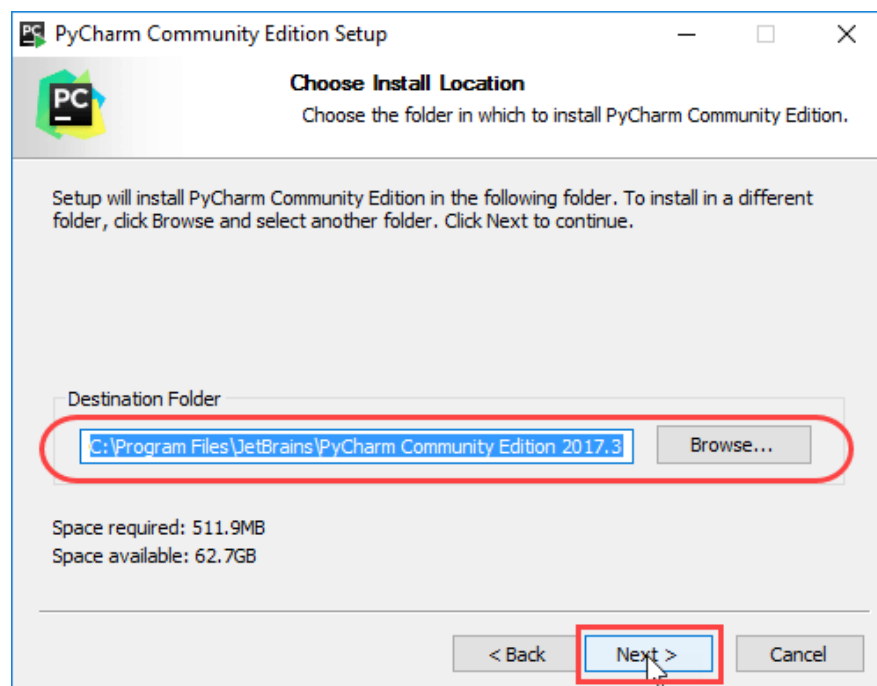
DOWNLOAD

Free, open-source

**Step 2)** Once the download is complete, run the exe for install PyCharm. The setup wizard should have started. Click “Next”.

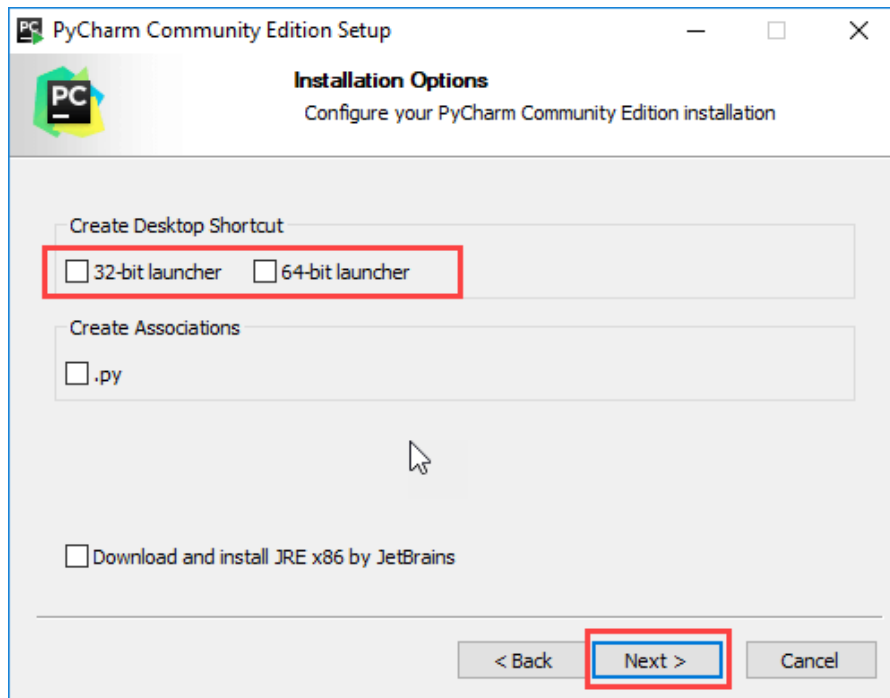


**Step 3)** On the next screen, Change the installation path if required. Click “Next”.

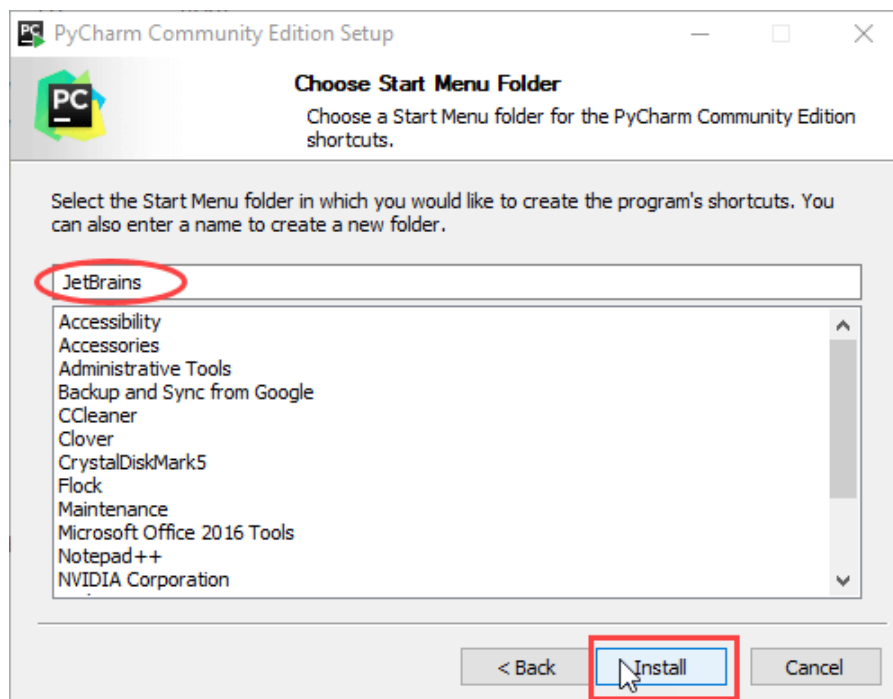


**Step 4)** On the next screen, you can create a desktop shortcut if you want and click on “Next”.

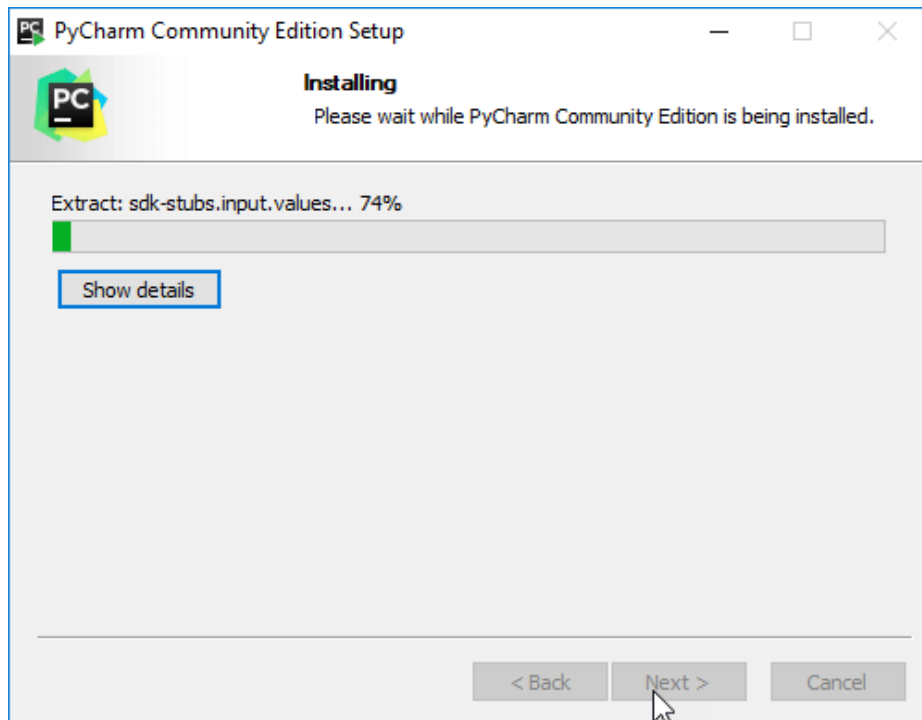




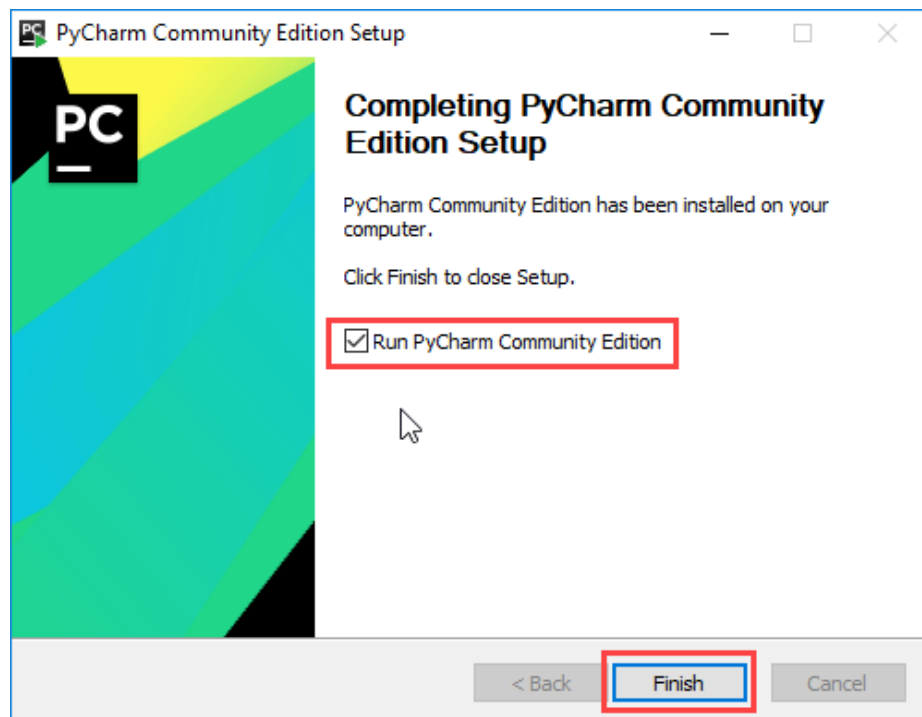
**Step 5)** Choose the start menu folder. Keep selected JetBrains and click on “Install”.



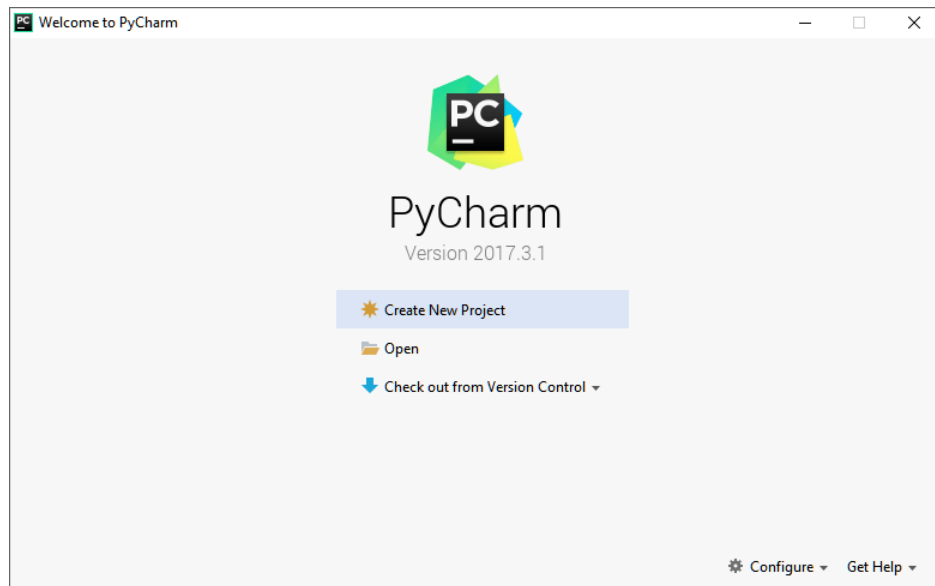
**Step 6)** Wait for the installation to finish.



**Step 7)** Once installation finished, you should receive a message screen that PyCharm is installed. If you want to go ahead and run it, click the “Run PyCharm Community Edition” box first and click “Finish”.

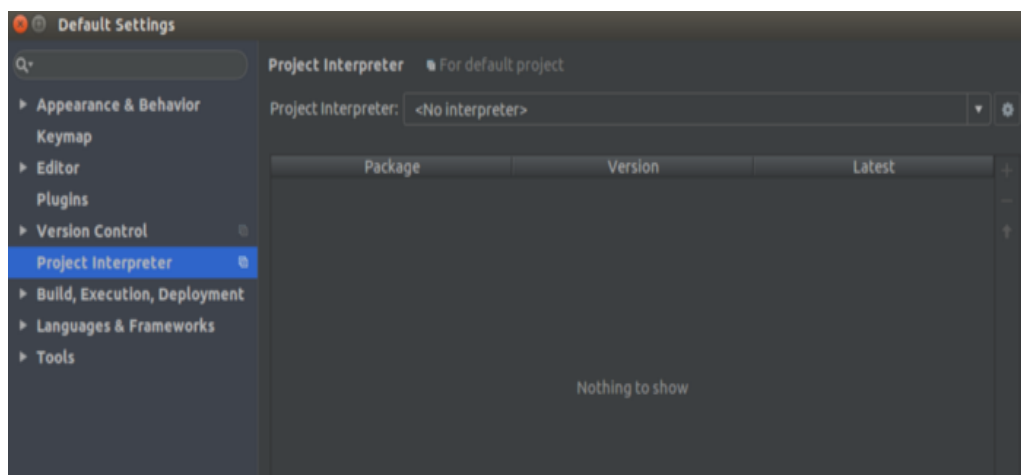


**Step 8)** After you click on “Finish,” the Following screen will appear.

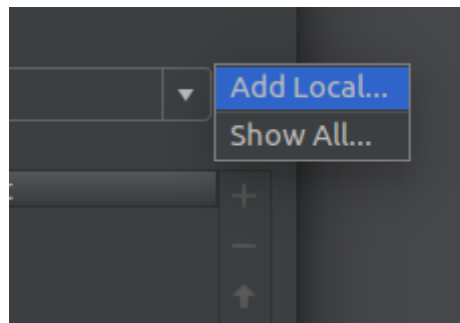


Click on Configure > Settings to open up settings in PyCharm

Search for "Project Interpreter". My PyCharm looks like this



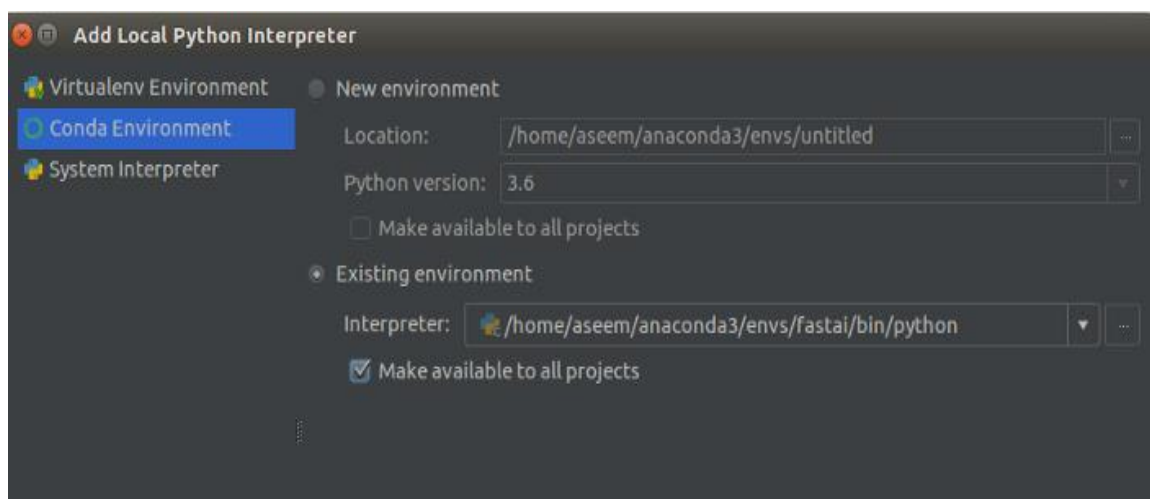
Click on Add local via the settings on the right side



Select “conda environment”

Click on “Existing environment” and navigate to the environment that you want to use. Note that you have to select the bin/python file inside the conda environment for PyCharm to be able to recognise the environment

Make sure to click the “Make available to all projects” if you want the interpreter to be used by multiple projects



Click ok and you are done

### 3.3 System Use case

Here is an example of a use case for a suicidal content detection system using NLP and machine learning techniques:

Use Case: Suicidal Content Detection for Social Media Platforms

Goal: To detect and prevent suicidal content on social media platforms

Actors: Social media users, moderators, mental health professionals

**Pre-conditions:**

- The system has been trained on a dataset of posts labeled as suicidal or non-suicidal
- The system has been integrated into the social media platform

**Flow of Events:**

- A social media user posts a message that contains language that suggests suicidal ideation
- The system analyzes the message using NLP techniques, such as sentiment analysis, keyword extraction, and/or topic modeling, to determine if the post is indicative of suicidal content
- If the system identifies the post as potentially suicidal, it sends an alert to the social media platform's moderators and/or mental health professionals
- The moderators and/or mental health professionals review the alert and take appropriate action, such as reaching out to the user to offer support and resources or removing the post if it violates the platform's community guidelines

**Post-conditions:**

- If the system identifies a post as suicidal, appropriate action is taken to prevent harm to the user and others

**Alternate Flow:**

- If the system does not identify a post as potentially suicidal, the post is published as usual and no alert is sent.

**Assumptions:**

- The system is trained on a representative dataset of posts and has high accuracy in detecting suicidal content
- The social media platform has agreed to integrate the system and has appropriate policies and procedures in place to handle alerts and take appropriate action

This use case illustrates how a suicidal content detection system using NLP and machine learning techniques can be integrated into social media platforms to detect and prevent potential harm to users. By identifying potentially suicidal posts, the system can alert moderators and mental health professionals to provide timely intervention and support.

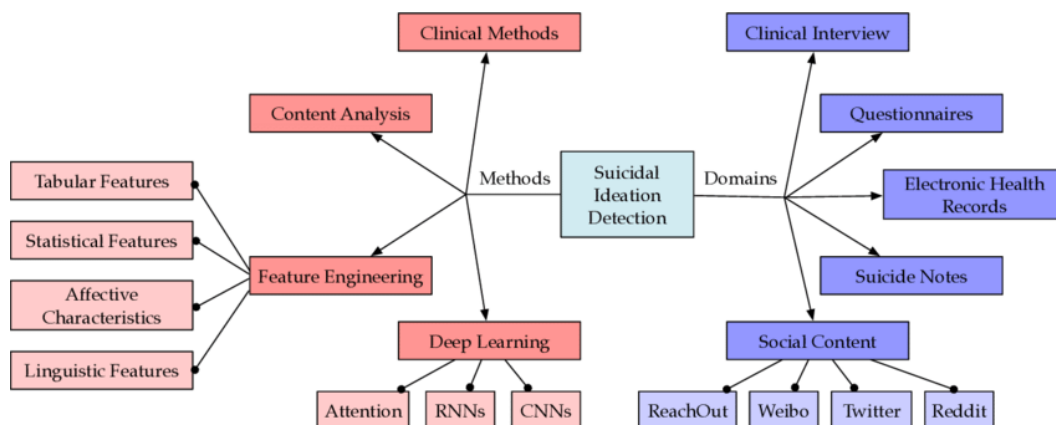


Fig: 3.1 System Use Case

## CHAPTER 4

## DESCRIPTION OF PROPOSED SYSTEM

The proposed system for suicidal content detection using NLP and machine learning techniques aims to identify social media posts that contain language indicative of suicidal ideation. The system utilizes a combination of natural language processing (NLP) techniques, such as sentiment analysis, keyword extraction, and topic modeling, along with machine learning algorithms to analyze social media posts and classify them as suicidal or non-suicidal.

### **The system will follow the following steps:**

- DATA COLLECTION –suicidal and non suicidal messages
- Nltk preprocessing – lemmatisation, stemming, emoji removal, #romoval, special characters removal, white space etc.
- Data modelling – logistic regression
- Data testing, precision recall, f1score
- Webscraping-beautiful soup
- Gui tkinter- user interface
- url submission and notification to parents – alert message

The proposed system aims to provide an effective solution to detect suicidal content on social media platforms, potentially saving lives by providing timely intervention and support to users. The system's accuracy and effectiveness will be tested through rigorous evaluation using relevant performance metrics. The system will also adhere to ethical and legal guidelines, ensuring the privacy and rights of users while preventing harm to individuals and society.

### **4.1 Selected Methodology or process model**

There are several process models available for developing software applications, such as the Waterfall model, the Agile model, and the Spiral model. In the case of developing a suicidal content detection system using NLP and machine learning techniques, an Agile methodology would be suitable due to the following reasons:

**Iterative approach:** Agile methodology follows an iterative approach, which involves breaking down the development process into small, manageable chunks. This approach is particularly useful for NLP and machine learning projects because it allows for continuous feedback and evaluation, enabling the team to refine and improve the system continuously.

**Flexibility:** The Agile methodology is flexible, which means that it can adapt to changes and updates in requirements quickly. In the case of developing a suicidal content detection system, it is essential to remain up-to-date with the latest research and developments in the field to ensure the system's accuracy and effectiveness.

**Collaboration:** Agile methodology promotes collaboration and communication between team members, stakeholders, and end-users. This collaboration ensures that the system's design and implementation meet the needs and expectations of all parties involved.

**Testing and quality assurance:** Agile methodology emphasizes the importance of testing and quality assurance throughout the development process. This approach is particularly important for developing a suicidal content detection system as it is essential to ensure that the system accurately identifies and handles potentially harmful content.

**Continuous improvement:** Agile methodology is focused on continuous improvement, allowing for the system to evolve and improve over time. This focus is particularly important for developing a suicidal content detection system as it enables the system to remain up-to-date with the latest research and developments, ensuring the system's effectiveness in identifying and preventing potentially harmful content.

Therefore, the Agile methodology would be an appropriate methodology for the



development of a suicidal content detection system using NLP and machine learning techniques.

## **4.2 Architecture / Overall Design of Proposed System**

The proposed system for suicidal content detection using NLP and machine learning techniques can be designed as follows:

**Data Collection:** The first step is to collect data from various sources such as social media platforms, forums, and blogs, where users may express suicidal thoughts. This data should be collected in a secure and ethical manner to protect the privacy of the users.

**Data Preprocessing:** Once the data is collected, it needs to be preprocessed to remove noise and irrelevant information. This involves text normalization, tokenization, stop-word removal, stemming, and lemmatization.

**Feature Extraction:** The next step is to extract relevant features from the preprocessed text. This can be done using techniques such as bag-of-words, TF-IDF, and word embeddings.

**Model Selection:** After feature extraction, the next step is to select an appropriate machine learning model. Several models can be evaluated, including logistic regression, decision trees, random forests, and neural networks, to determine the most accurate model.

**Training and Testing:** The selected model needs to be trained on the labeled data. The labeled data contains instances of suicidal and non-suicidal content. Once the model is trained, it needs to be tested on new, unseen data to evaluate its performance.

**Model Deployment:** After the model has been tested and optimized, it can be deployed as an API or web service. Users can then interact with the system by

submitting text data, and the model will provide a prediction of whether the text contains suicidal content or not.

**Continuous Improvement:** The system should be continuously monitored and evaluated to ensure that it is accurate and up-to-date. As new data becomes available, the model can be retrained to improve its performance. Additionally, user feedback can be incorporated to improve the system's accuracy and usability.

Overall, the system should be designed to provide an accurate, reliable, and user-friendly solution for detecting suicidal content in text data. The architecture should incorporate the latest NLP and machine learning techniques to ensure that the system is efficient, scalable, and effective.

## **BLOCK DIAGRAM**

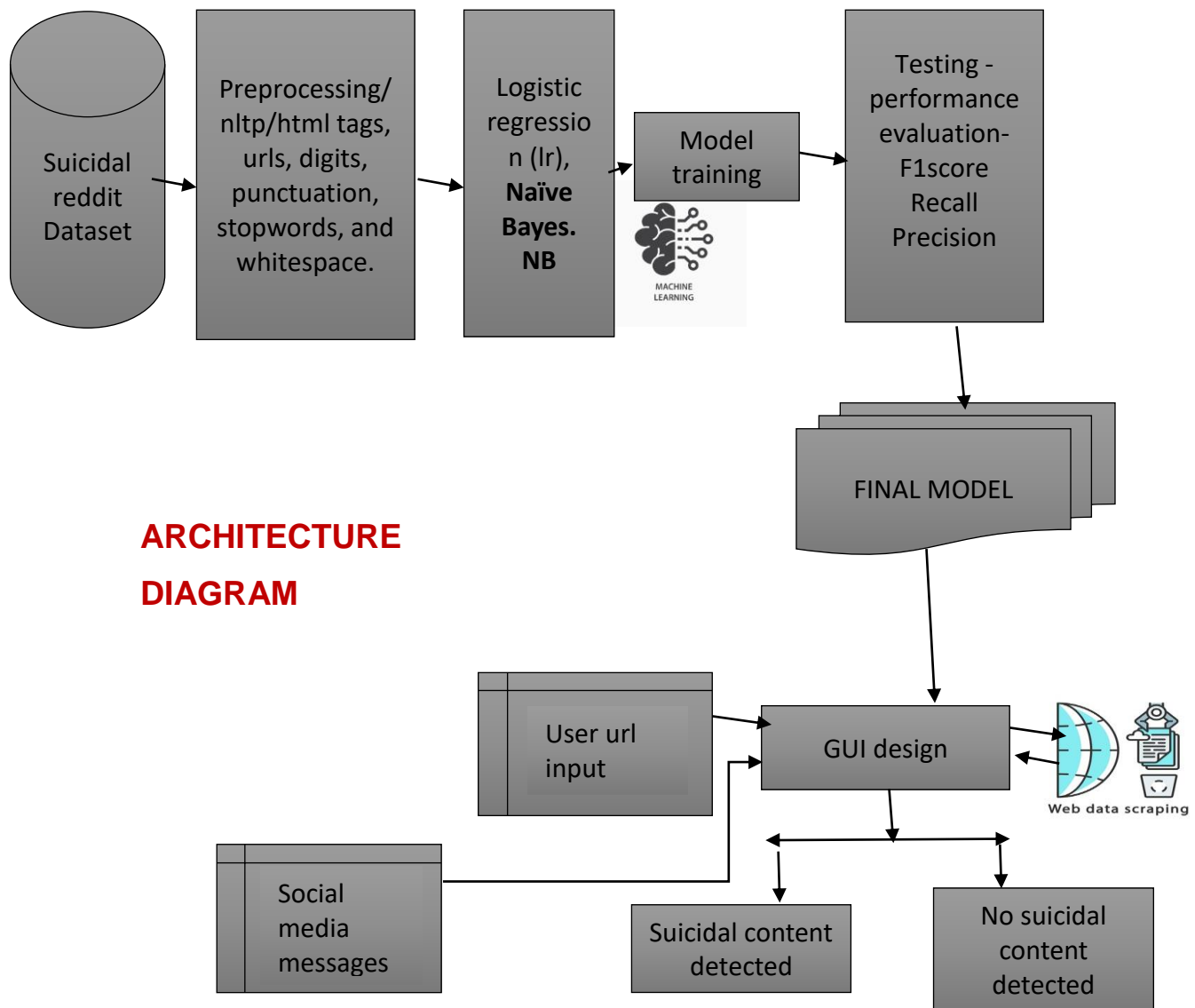


Fig: 4.1 Architecture Diagram

## FLOW DIAGRAM

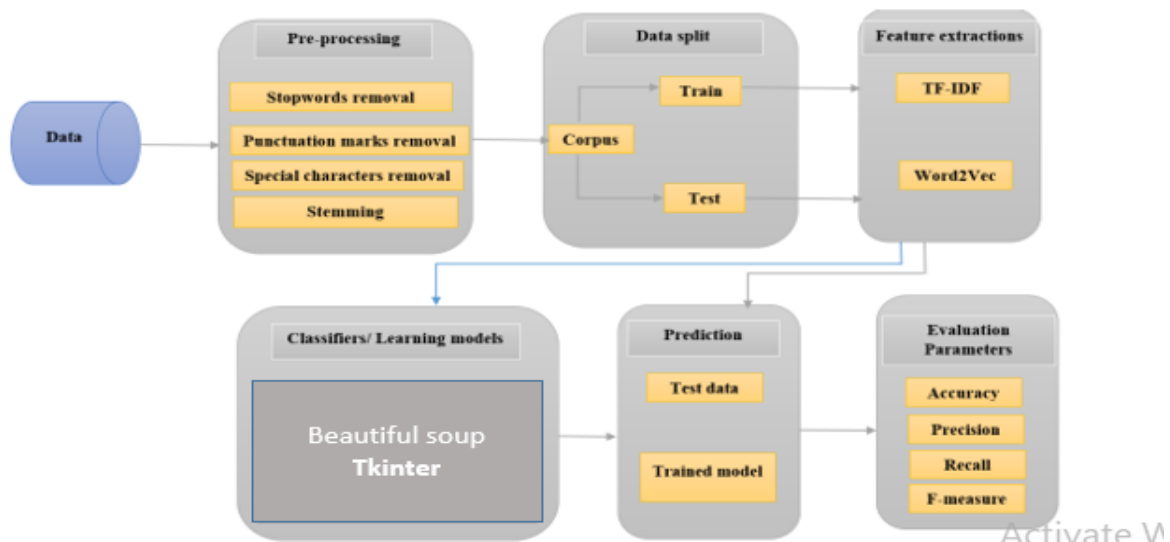


Fig: 4.2 Flow Diagram

### 4.3 Description of Software for Implementation and Testing plan of the Proposed Model/System

#### Description of Software for Implementation:

The software for implementing the proposed suicidal content detection system can be developed using programming languages such as Python or Java. The following libraries and frameworks can be used for different tasks:

**Data Collection:** Python libraries such as Beautiful Soup, Selenium, and Scrapy can be used to collect data from various sources.

**Data Preprocessing:** Libraries such as NLTK, Spacy, and TextBlob can be used to preprocess the collected data.

**Feature Extraction:** Libraries such as Scikit-learn, Gensim, and TensorFlow can be used to extract features from the preprocessed data.

**Model Selection:** Different machine learning models can be evaluated using libraries such as Scikit-learn and TensorFlow.

**Model Deployment:** The model can be deployed as a web service using frameworks such as Flask or Django.

### **Testing Plan of the Proposed Model/System:**

The testing plan for the proposed suicidal content detection system can be divided into the following steps:

**Data Preparation:** A labeled dataset of suicidal and non-suicidal content needs to be prepared for training and testing the model.

**Model Training:** The selected machine learning model needs to be trained on the labeled dataset.

**Model Evaluation:** The trained model needs to be evaluated on a separate test dataset to determine its accuracy, precision, recall, and F1 score.

**Performance Tuning:** If the model's performance is not satisfactory, hyperparameter tuning and feature selection can be performed to improve its accuracy.

**User Testing:** Once the model is optimized, it needs to be tested by real users to ensure that it is accurate and user-friendly.

**Continuous Improvement:** The system should be continuously monitored and evaluated to ensure that it is up-to-date and accurate. New data can be used to retrain the model and improve its performance.

Overall, the testing plan should ensure that the proposed system is accurate,

reliable, and user-friendly. It should also be scalable and adaptable to new data and user feedback.

#### **4.4 Project Management Plan**

The project management plan for the proposed suicidal content detection system using NLP and machine learning techniques can be structured as follows:

**Project Scope:** The scope of the project includes developing a system that can accurately detect suicidal content in text data using NLP and machine learning techniques. The system should be user-friendly and scalable.

**Project Timeline:** The project timeline should be divided into different phases, including data collection, data preprocessing, feature extraction, model selection, model training, model evaluation, model deployment, and continuous improvement. Each phase should have specific milestones and deadlines.

**Resource Allocation:** The project team should include data scientists, machine learning experts, NLP experts, and software developers. The team members should be allocated to different phases of the project based on their expertise.

**Risk Management:** Potential risks and challenges should be identified, and contingency plans should be developed to mitigate them. Risks can include data privacy and security concerns, lack of labeled data, overfitting, and underfitting.

**Communication Plan:** The project team should establish a communication plan to ensure that all team members are informed about the project's progress, challenges, and milestones. Regular meetings and progress reports should be scheduled.

**Quality Assurance:** The system should be tested at each phase to ensure that it meets the project's requirements and specifications. Quality assurance should be performed by a dedicated team that is independent of the development team.

**Budget and Funding:** The project budget should include costs for data collection,

software development, hardware, and personnel. Funding can be obtained from government grants, private organizations, or crowdfunding.

**Project Monitoring and Evaluation:** The project should be monitored and evaluated at each phase to ensure that it is progressing according to the timeline and budget. Key performance indicators should be defined to measure the system's accuracy, usability, and scalability.

Overall, the project management plan should ensure that the proposed system is developed within the allocated timeline, budget, and scope. The plan should also ensure that the system meets the project's requirements and specifications and is of high quality.

#### **4.5 Financial report on estimated costing**

The estimated costing for the development of a suicidal content detection system using NLP and machine learning techniques can vary depending on several factors, such as the complexity of the system, the amount of data required, and the size of the development team. However, a general estimation of the costs involved is as follows:

**Data Collection:** The cost of data collection can vary depending on the sources used. Social media platforms and forums can be used for collecting data, which may be free or require a subscription. The cost for data collection can range from \$500 to \$5,000.

**Infrastructure:** The cost of infrastructure required for developing the system can include hardware, software, and cloud-based services. The cost for hardware can range from \$1,000 to \$10,000, and the cost for software can range from \$500 to \$5,000. Cloud-based services can also be used, which may cost around \$500 to \$1,000 per month.

**Personnel:** The development team may include data scientists, machine learning experts, NLP experts, and software developers. The cost of personnel can vary

depending on their expertise and location. For example, the average salary for a data scientist in the US is around \$120,000 per year, while the average salary for a software developer is around \$90,000 per year. The cost for personnel can range from \$50,000 to \$200,000.

**Miscellaneous Expenses:** Miscellaneous expenses can include costs for project management, quality assurance, and contingency planning. The cost for miscellaneous expenses can range from \$5,000 to \$10,000.

Overall, the estimated costing for developing a suicidal content detection system using NLP and machine learning techniques can range from \$50,000 to \$250,000, depending on the factors mentioned above. It is essential to note that these costs are estimates and may vary based on the project's specific requirements and scope.

#### **4.6 Transition/ Software to Operations Plan**

Transitioning the suicidal content detection system from development to operations involves several steps to ensure a smooth and successful transition. The following are the key steps in the software to operations plan:

**Finalizing the System:** The system should be finalized, tested, and validated before transitioning it to operations. The system should be evaluated for accuracy, scalability, and usability. Any issues or bugs should be addressed before the transition.

**Documentation:** The system should be documented, including the system architecture, design, and functionality. The documentation should be comprehensive and easy to understand, and it should be made available to the operations team.

**Deployment:** The system should be deployed to the production environment. The deployment process should be thoroughly tested to ensure that the system is deployed correctly.



**Training:** The operations team should be trained on how to use the system, including its features, functionality, and maintenance. The training should be comprehensive and include troubleshooting procedures.

**Maintenance and Support:** The operations team should be responsible for maintaining and supporting the system. The team should be available to address any issues that arise, such as system failures, errors, and user requests.

**Monitoring:** The system should be monitored regularly to ensure that it is performing as expected. Monitoring should include system logs, user feedback, and performance metrics. Any issues should be addressed promptly to minimize disruptions.

**Continuous Improvement:** The system should be continuously improved based on feedback from users and monitoring results. Improvements can include adding new features, optimizing the system's performance, and improving its accuracy.

Overall, the transition from software development to operations requires careful planning and coordination to ensure that the system is deployed successfully and operates efficiently. The transition plan should be comprehensive and include all necessary steps, such as finalizing the system, documentation, deployment, training, maintenance and support, monitoring, and continuous improvement.

## CHAPTER 5

### IMPLEMENTATION DETAILS

The implementation of a suicidal content detection system using NLP and machine learning techniques involves several steps, including data collection, preprocessing, feature extraction, model selection, and evaluation. The following are the implementation details for each of these steps:

**Data Collection:** The first step is to collect data from various sources, including social media platforms, forums, and websites. The data should be labeled as either suicidal or non-suicidal.

**Preprocessing:** The collected data needs to be preprocessed to remove noise, stop words, and irrelevant information. The preprocessing step involves tokenization, stemming, and lemmatization.

**Feature Extraction:** The next step is to extract features from the preprocessed data. The features can include word frequency, TF-IDF scores, and sentiment analysis.

**Model Selection:** Several machine learning models can be used for suicidal content detection, including Naive Bayes, Support Vector Machines (SVM), and Random Forests. The model should be selected based on its performance and accuracy.

**Model Training:** The selected model should be trained on the labeled data using cross-validation techniques.

**Model Evaluation:** The trained model should be evaluated using evaluation metrics such as precision, recall, and F1 score. The model should be tested on a separate dataset to ensure that it is not overfitting.

**Deployment:** The final step is to deploy the model into the production environment. The model can be integrated into existing applications, or a new application can be developed specifically for suicidal content detection.

The implementation of a suicidal content detection system using NLP and machine learning techniques requires a skilled team of data scientists, machine learning experts, and software developers. The team should have experience in natural language processing, machine learning, and software development. The implementation should be done in an iterative manner, with regular testing and evaluation to ensure that the system is accurate and efficient.

## 5.1 Development and Deployment Setup

The development and deployment setup for a suicidal content detection system using NLP and machine learning techniques involves several components, including hardware, software, and tools. The following are the implementation details for each of these components:

**Hardware:** The hardware required for development and deployment depends on the size and complexity of the system. A typical setup includes a powerful computer or server with sufficient memory and processing power to handle large datasets.

**Software:** The software required for development and deployment includes programming languages, libraries, and frameworks. The following are the most commonly used software tools for suicidal content detection using NLP and machine learning techniques:

- **Programming Languages:** Python, Java, R

## **Libraries:**

### **Beautiful soup:**

- Beautiful Soup is a Python library that is used for web scraping purposes to pull the data out of HTML and XML files. It creates a parse tree from page source code that can be used to extract data in a hierarchical and more readable manner.
- It is a complete framework for web-scraping or crawling. Beautiful Soup is a parsing library which also does a pretty good job of fetching contents from URL and allows you to parse certain parts of them without any hassle. It only fetches the contents of the URL that you give and then stops.
- Web scraping or Web data extraction is a method to collect data from websites. ... Most of this data is unstructured data in an HTML format which is then converted into structured data in a spreadsheet or a database so that it can be used in various applications.
- Comparing selenium vs Beautiful Soup allows you to see that Beautiful Soup is more user-friendly and allows you to learn faster and begin web scraping smaller tasks easier. Selenium on the other hand is important when the target website has a lot of java elements in its code.

### **Tkinter:**

- Tkinter is a library written in Python that is widely used to create GUI applications. It is very easy to build GUI using Tkinter and the process is even faster. Tkinter has several widgets that can be used while developing GUI. These include buttons, radio buttons, checkboxes, etc
- ML requires continuous data processing, and Python's libraries let you access, handle and transform data. These are some of the most widespread libraries you can use for ML and AI: Scikit-learn for handling basic ML

algorithms like clustering, linear and logistic regressions, regression, classification, and others.

- Tkinter offers access to geometric configuration of the widgets which organize the widgets in parent windows in python for data science. Mainly three geometry manager classes are there.
- Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit. ... Add one or more of the above-mentioned widgets to the GUI application.
- If your goal is to learn how to create GUIs, tkinter is arguably one of the best toolkits there is to reach that goal. It's simple and easy to learn, and can provide a fantastic introduction to concepts you must master in order to create graphical desktop applications.

**Frameworks:** Flask, Django

**Tools:** The tools required for development and deployment include text editors, development environments, and deployment platforms. The following are the most commonly used tools for suicidal content detection using NLP and machine learning techniques:

**Text Editors:** Visual Studio Code, PyCharm, Sublime Text

**Development Environments:** Anaconda, Jupyter Notebook, Spyder

**Deployment Platforms:** Amazon Web Services (AWS), Google Cloud Platform (GCP), Microsoft Azure

The development and deployment setup for a suicidal content detection system should be designed with scalability, reliability, and security in mind. The system should be tested thoroughly before deployment to ensure that it meets the requirements and performs as expected. The deployment process should be automated to minimize errors and reduce downtime. Finally, the system should be

monitored continuously to ensure that it is running smoothly and to detect any issues early on.

## 5.2 Algorithms

### 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 dependant 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.

Many people may have a question, whether Logistic Regression is a classification or regression category. The logistic regression hypothesis suggests that the cost function be limited to a value between 0 and 1. As a result, linear functions fail to describe it since it might have a value larger than 1 or less than 0, which is impossible according to the logistic regression hypothesis.

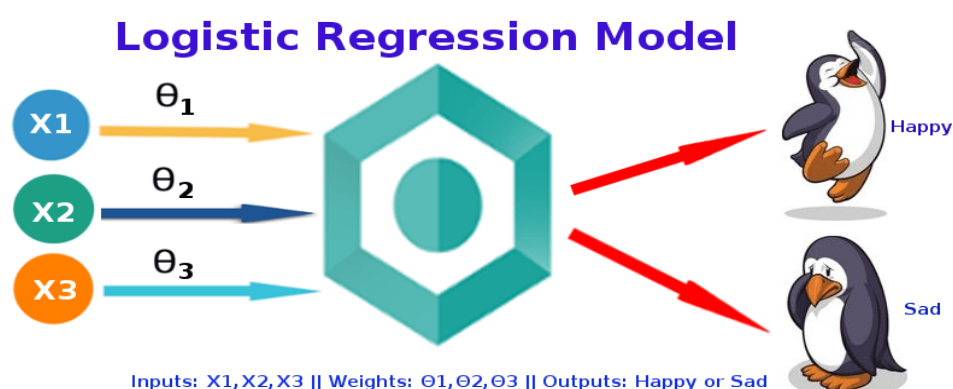


Fig: 5.1 Logistic Regression Model

To answer this question, Logistic Regression is considered a regression model also. This model creates a regression model to predict the likelihood that a given data entry belongs to the category labeled “1.” Logistic regression models the data using the sigmoid function, much as linear regression assumes that the data follows a linear distribution.

Why the name Logistic Regression?

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.

## Tkinter

- Tkinter is a library written in Python that is widely used to create GUI applications. It is very easy to build GUI using Tkinter and the process is even

faster. Tkinter has several widgets that can be used while developing GUI. These include buttons, radio buttons, checkboxes, etc

- ML requires continuous data processing, and Python's libraries let you access, handle and transform data. These are some of the most widespread libraries you can use for ML and AI: Scikit-learn for handling basic ML algorithms like clustering, linear and logistic regressions, regression, classification, and others.
- tkinter offers access to geometric configuration of the widgets which organize the widgets in parent windows in python for data science. Mainly three geometry manager classes class are there.
- Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit. ... Add one or more of the above-mentioned widgets to the GUI application.
- If your goal is to learn how to create GUIs, tkinter is arguably one of the best toolkits there is to reach that goal. It's simple and easy to learn, and can provide a fantastic introduction to concepts you must master in order to create graphical desktop applications.

## **MACHINE LEARNING**

### **DEFINITION:**

Machine Learning is a category of algorithms that allow software applications to predict much better results without being specifically programmed. The basic premise of machine learning is to build algorithms that receive input data and use statistical analysis to predict output data while output data is updated like many input data become valid. The processes involved in machine learning are similar to the processes of data mining and predictive modelling. Both require searching for certain patterns by date, and adjusting program actions accordingly. Many people are also familiar with machine learning from internet shopping and the advertisements that are shown to them depending on what they are buying. This is because referral engines use machine learning to customize ads that are delivered online in near real time. In addition to personalized marketing, other well-known cases in which



machine learning is used are fraud detection, spam filtering, threat detection of countries in the network, maintenance, predictability, and building the flow of news.

## HOW MACHINE LEARNING WORKS:

Machine learning algorithms are categorized as both supervised and unsupervised.

### Supervised Algorithms

They require a data researcher, or data analyst, who has the knowledge of machine learning to supply the desired input and output data, in addition to delivering feedback on the accuracy of the predictions; acute during algorithm training. Data researchers determine which variables, or characteristics, should be analyzed by the model and used to develop predictions. Once the training is complete, the algorithm will apply what it has learned to new data. Supervised learning problems can be further grouped into regression and classification problems. Classification: A classification problem is when the output variable is a category, such as “red” or “blue” or “disease” and “no disease”. Regression: A regression problem is when the output variable is a real value, such as “dollars” or “weight”. Some common types of problems built on top of classification and regression include recommendation and time series prediction respectively. Some popular examples of supervised machine learning algorithms are: Linear regression for regression problems. Random forest for classification and regression problems, Support vector machines for classification problems.

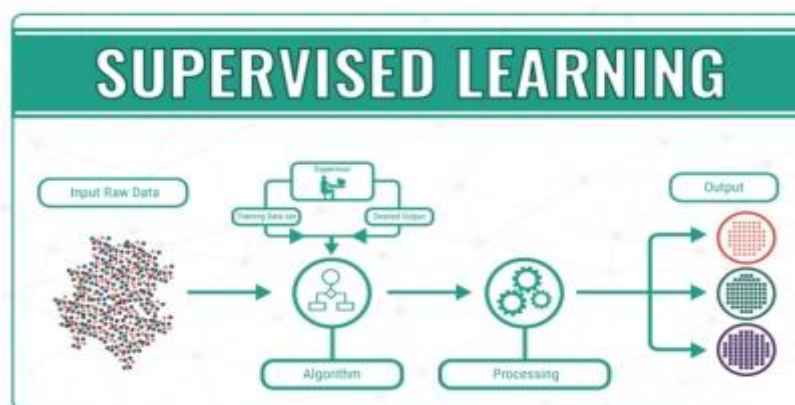


Fig: 5.2 Supervised learning

## **Unsupervised Algorithms**

They do not need training with output data. Instead, they use a method called deep learning to review the data and come to conclusions. Unsupervised and learned algorithms, also known as neural networks, are used for more complex processes than supervised algorithms, which include image recognition, speech-to-text, and natural language generation. These neural networks work by first combining millions of training examples with data and automatically identifying subtle correlations between multiple variables. Once trained, the algorithm can be used by associates to interpret new data. These algorithms become feasible only in the information age, because they require massive amounts of data to train. These are called unsupervised learning because unlike supervised learning above there is no correct answers and there is no teacher. Algorithms are left to their own devices to discover and present the interesting structure in the data. Unsupervised learning problems can be further grouped into clustering and association problems. Clustering: A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behavior. Association: An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y. Some popular examples of unsupervised learning algorithms are: k-means for clustering problems. Apriori algorithm for association rule learning problems.

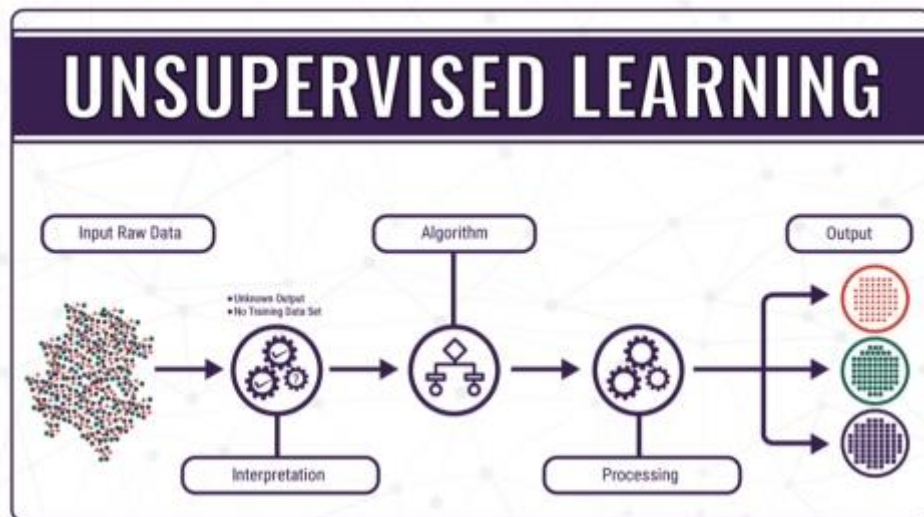


Fig: 5.3 Unsupervised Algorithms

## RANDOM FOREST

- Random Forest algorithm is derived from the random tree, which is a type of decision tree. Therefore, the first element discussed will be the Decision Tree.
- The Decision Tree creates a hierarchical division of data from the set, where a homogeneous division into classes is obtained at the tree leaf level.
- Each vertex corresponds to the selected attribute describing the instances in the set, and the edges speak about the set of values of individual attributes.
- The tree structure is usually top-down, i.e. from the root to the leaves

## K-NEAREST NEIGHBOUR

- K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.

## **DECISION TREE**

- Decision Trees are a type of Supervised Machine Learning (that is you explain what the input is and what the corresponding output is in the training data) where the data is continuously split according to a certain parameter.
- The tree can be explained by two entities, namely decision nodes and leaves.

## **SVM**

- Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

## **NAIVE BAYES**

- Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.

## **5.3 Testing**

### **SYSTEM TESTING**

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example the design must not have any logic faults in the design is detected before coding commences, otherwise the cost of fixing the faults will be

considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as walkthrough.

Testing is one of the important steps in the software development phase. Testing checks for the errors, as a whole of the project testing involves the following test cases:

- Static analysis is used to investigate the structural properties of the Source code.
- Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

## **TEST DATA AND OUTPUT**

- **UNIT TESTING**

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module. The white-box testing techniques were heavily employed for unit testing.

- **FUNCTIONAL TESTS**

Functional test cases involved exercising the code with nominal input values for which the expected results are known, as well as boundary values and special values, such as logically related inputs, files of identical elements, and empty files.

Three types of tests in Functional test:

- ✓ Performance Test
- ✓ Stress Test
- ✓ Structure Test

## **INTEGRATION TESTING**

Integration testing is a systematic technique for construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. i.e., integration testing is the complete testing of the set of modules which makes up the product. The objective is to take untested modules and build a program structure tester should identify critical modules. Critical modules should be tested as early as possible. One approach is to wait until all the units have passed testing, and then combine them and then tested. This approach is evolved from unstructured testing of small programs. Another strategy is to construct the product in increments of tested units. A small set of modules are integrated together and tested, to which another module is added and tested in combination. And so on. The advantages of this approach are that, interface dispenses can be easily found and corrected.

The major error that was faced during the project is linking error. When all the modules are combined the link is not set properly with all support files. Then we checked out for interconnection and the links. Errors are localized to the new module and its intercommunications. The product development can be staged, and modules integrated in as they complete unit testing. Testing is completed when the last module is integrated and tested.

## **TESTING TECHNIQUES / TESTING STRATERGIES**

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet –undiscovered error. A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It

verifies that the whole set of programs hang together. System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing.

The software testing process commences once the program is created and the documentation and related data structures are designed. Software testing is essential for correcting errors. Otherwise the program or the project is not said to be complete. Software testing is the critical element of software quality assurance and represents the ultimate the review of specification design and coding. Testing is the process of executing the program with the intent of finding the error. A good test case design is one that as a probability of finding an yet undiscovered error. A successful test is one that uncovers an yet undiscovered error. Any engineering product can be tested in one of the two ways:

- **WHITE BOX TESTING**

This testing is also called as Glass box testing. In this testing, by knowing the specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

Basis path testing:

- ✓ Flow graph notation
- ✓ Cyclometric complexity
- ✓ Deriving test cases
- ✓ Graph matrices Control

- **BLACK BOX TESTING**

In this testing by knowing the internal operation of a product, test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

The steps involved in black box test case design are:

- ✓ Graph based testing methods
- ✓ Equivalence partitioning
- ✓ Boundary value analysis
- ✓ Comparison testing
- ✓

### **PROGRAM TESTING:**

The logical and syntax errors have been pointed out by program testing. A syntax error is an error in a program statement that in violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted keywords are common syntax error. These errors are shown through error messages generated by the computer. A logic error on the other hand deals with the incorrect data fields, out-off-range items and invalid combinations. Since the compiler s will not deduct logical error, the programmer must examine the output. Condition testing exercises the logical conditions contained in a module. The possible types of elements in a condition include a Boolean operator, Boolean variable, a pair of Boolean parentheses A relational operator or on arithmetic expression. Condition testing method focuses on testing each condition in the program the purpose of condition test is to deduct not only errors in the condition of a program but also other a errors in the program.

### **SECURITY TESTING:**

Security testing attempts to verify the protection mechanisms built in to a system



well, in fact, protect it from improper penetration. The system security must be tested for invulnerability from frontal attack must also be tested for invulnerability from rear attack. During security, the tester places the role of individual who desires to penetrate system.

### **VALIDATION TESTING:**

At the culmination of integration testing, software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test-validation testing begins. Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in manner that is reasonably expected by the customer. Software validation is achieved through a series of black box tests that demonstrate conformity with requirement. After validation test has been conducted, one of two conditions exists.

- \* The function or performance characteristics confirm to specifications and are accepted.
- \* A validation from specification is uncovered and a deficiency created.

Deviation or errors discovered at this step in this project is corrected prior to completion of the project with the help of the user by negotiating to establish a method for resolving deficiencies. Thus the proposed system under consideration has been tested by using validation testing and found to be working satisfactorily. Though there were deficiencies in the system they were not catastrophic.

### **USER ACCEPTANCE TESTING:**

User acceptance of the system is key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system and user at the time of developing and making changes whenever required. This is done in regarding to the following points.

- Input screen design.
- Output screen design.

## CHAPTER 6

### RESULT AND DISCUSSION

The result and discussion of a suicidal content detection system using NLP and machine learning techniques depend on several factors, such as the quality of the data, the choice of algorithm, and the evaluation metrics used. Here are some potential results and discussion points:

**Model Performance:** The performance of the model can be evaluated using metrics like accuracy, precision, recall, and F1 score. A high accuracy and F1 score indicate that the model is effective at detecting suicidal content. It is important to compare the performance of the model with other state-of-the-art models and establish a benchmark for future improvements.

**Feature Importance:** Understanding the importance of specific features in the model can provide insight into the language and patterns used in suicidal content. It can also inform the development of targeted interventions and prevention strategies.

**Limitations and Challenges:** Suicidal content detection systems face several challenges and limitations, such as the difficulty of accurately detecting sarcasm or irony, the potential for false positives and false negatives, and the need for ongoing evaluation and improvement. It is important to acknowledge and address these limitations to ensure the system is used in a responsible and effective way.

**Ethical Considerations:** Suicidal content detection systems raise ethical considerations, such as privacy, confidentiality, and potential harm to individuals who may be flagged as at-risk for suicide. It is important to consider these ethical considerations and work with mental health professionals to ensure the system is used in a responsible and beneficial way.

Overall, the result and discussion of a suicidal content detection system using NLP and machine learning techniques can provide valuable insights into the language and patterns used in suicidal content, as well as the challenges and limitations of developing such a system. It is important to continuously evaluate and improve the system to ensure its effectiveness and ethical use.

## **CHAPTER 7**

### **7.1 Conclusion**

In conclusion, suicidal content detection using NLP and machine learning techniques has the potential to provide an effective and efficient tool for identifying at-risk individuals and preventing suicide. However, the development and deployment of such a system require careful consideration of technical and ethical factors, including data collection, feature selection, algorithm choice, evaluation metrics, deployment, and ethical considerations. The result and discussion of the system can provide valuable insights into the language and patterns used in suicidal content, as well as the challenges and limitations of developing such a system. Overall, suicidal content detection systems using NLP and machine learning techniques can be a valuable tool in suicide prevention efforts when developed and used in a

responsible and ethical way, in collaboration with mental health professionals and other experts in the field.

## 7.2 Future Work

There are several avenues for future work in the field of suicidal content detection using NLP and machine learning techniques. Here are some potential areas of focus:

**Multilingual Support:** Developing a suicidal content detection system that supports multiple languages can improve its effectiveness and reach a wider population.

**Fine-grained Detection:** Developing a system that can detect the severity and urgency of suicidal content can help prioritize interventions and resources for individuals at higher risk.

**Real-time Detection:** Developing a system that can detect suicidal content in real-time, such as in social media or online forums, can enable timely interventions and support.

**Incorporating Contextual Information:** Incorporating contextual information, such as demographic data or previous mental health history, can improve the accuracy of the model and personalize interventions.

**Addressing Ethical Concerns:** Addressing ethical concerns, such as privacy, confidentiality, and potential harm to individuals who may be flagged as at-risk for suicide, can improve the responsible and ethical use of the system.

**Collaborative Research:** Collaborating with mental health professionals and other experts in the field can improve the understanding of suicidal behavior and inform the development of

effective prevention strategies.

Overall, future work in the field of suicidal content detection using NLP and machine learning techniques can focus on improving the effectiveness, efficiency, and ethical use of the system to prevent suicide and support at-risk individuals.

### **7.3 Research Issues**

There are several research issues related to suicidal content detection using NLP and machine learning techniques. Here are some potential areas of focus:

**Data Availability and Quality:** The availability and quality of data are critical to the development and evaluation of suicidal content detection systems. However, collecting and annotating data related to suicide can be challenging due to ethical, legal, and social concerns. Future research can focus on addressing these challenges and improving the quality and quantity of data available for the development and evaluation of such systems.

**Algorithm Selection and Evaluation:** Choosing an appropriate algorithm and evaluation metrics are critical to the performance and effectiveness of suicidal content detection systems. However, the choice of algorithm and evaluation metrics can be subjective and depend on various factors, such as the data and the context of use. Future research can focus on developing standardized evaluation metrics and comparing the performance of different algorithms across various contexts.

**Interpretability and Explainability:** The interpretability and explainability of suicidal content detection systems are critical to gaining the trust and acceptance of mental health

professionals and the general public. However, many machine learning algorithms, such as deep learning, can be opaque and difficult to interpret. Future research can focus on developing more interpretable and explainable machine learning models and providing insights into the language and patterns used in suicidal content.

**Addressing Bias and Fairness:** Suicidal content detection systems can be prone to bias and unfairness, especially when trained on imbalanced or unrepresentative data. Future research can focus on addressing these issues and developing techniques to ensure the fairness and equity of such systems.

**Ethical Considerations:** Suicidal content detection systems raise several ethical considerations, such as privacy, confidentiality, and potential harm to individuals who may be flagged as at-risk for suicide. Future research can focus on developing ethical frameworks and guidelines for the development and use of such systems.

Overall, addressing these research issues can improve the development, evaluation, and responsible use of suicidal content detection systems using NLP and machine learning techniques.

## **7.4 Implementation Issues**

There are several implementation issues related to suicidal content detection using NLP and machine learning techniques. Here are some potential areas of focus:

**Data Privacy and Security:** Data privacy and security are critical concerns when developing and deploying suicidal content detection systems. It is essential to ensure that sensitive

information, such as personal and health data, is protected from unauthorized access and use.

**Integration with Existing Systems:** Suicidal content detection systems can be more effective when integrated with existing mental health systems, such as crisis helplines, hotlines, or chatbots. However, integrating these systems can be challenging due to technical, organizational, and regulatory factors.

**User Acceptance and Adoption:** The success of suicidal content detection systems depends on the acceptance and adoption by mental health professionals, individuals at risk of suicide, and the general public. Ensuring that the system is user-friendly, effective, and ethical can improve the acceptance and adoption of the system.

**Scalability and Sustainability:** Suicidal content detection systems need to be scalable and sustainable to accommodate the increasing demand for mental health support and services. Ensuring that the system can handle a large volume of data and users, and that it is cost-effective and sustainable in the long term, can improve its scalability and sustainability.

**Collaboration with Mental Health Professionals:** Collaborating with mental health professionals and experts in the field can improve the development, evaluation, and use of suicidal content detection systems. It can also ensure that the system aligns with the best practices and standards in mental health care.



Overall, addressing these implementation issues can improve the development, deployment, and responsible use of suicidal content detection systems using NLP and machine learning techniques.

## REFERENCE PAPERS

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».” [https://www.rospotrebnadzor.ru/documents/details.php?ELEMENT\\_ID=6735](https://www.rospotrebnadzor.ru/documents/details.php?ELEMENT_ID=6735) (accessed Nov. 24, 2020).
10. “О деятельности Роспотребнадзора по предотвращению самоубийств среди детей и подростков - RSS - Официальный сайт Роспотребнадзора.” [http://11.rospotrebnadzor.ru/rss\\_all/-/asset\\_publisher/Kq6J/content/id/382348](http://11.rospotrebnadzor.ru/rss_all/-/asset_publisher/Kq6J/content/id/382348) (accessed Nov. 24, 2020).
11. “Число детских suicides в России в 2016 году выросло почти на 60%.” <https://www.interfax.ru/russia/554375> (accessed Nov. 24, 2020).
12. 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.
13. 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.
14. “Tor Project | Anonymity Online.” <https://www.torproject.org/> (accessed Nov. 27, 2020).
15. 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.
16. 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.

- 17.B. Feinstein, D. Peck, and I. SecureWorks, "Caffeine monkey: Automated collection, detection and analysis of malicious javascript," *Black Hat USA*, vol. 2007, 2007.
- 18.C. N. Dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts.," in *COLING*, pp. 69–78, 2014.
- 19.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.
- 20.J. Saxe and K. Berlin, "Deep neural network based malware detection using two dimensional binary program features," in *Malicious and Unwanted Software (MALWARE), 2015 10th International Conference on*, pp. 11–20, IEEE, 2015.

## APPENDIX

### A. SOURCE CODE

```
import json
import tkinter as tk
import tkinter.scrolledtext as scrolledtext
import tkinter.ttk as ttk
from tkinter.filedialog import askopenfile

import requests
from bs4 import BeautifulSoup
from PIL import Image, ImageTk
from ttkthemes import ThemedStyle
from tkinter import messagebox

# 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()

class Startpage(ttk.Frame):

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

        label = ttk.Label(self, text="Detection of suicidal content", font=("Simplifica", 22))
        label.pack(pady=5, padx=5)

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

        button1 = ttk.Button(self, text="Detect",
                             command=lambda: controller.show_frame(PageOne))
        button1.pack()

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

        button2 = ttk.Button(self, text="About",
                             command=lambda: controller.show_frame(PageTwo))
        button2.pack()

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

```

```

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

```

```

class PageOne(ttk.Frame):

```

```

    def __init__(self, parent, controller):
        ttk.Frame.__init__(self, parent)
        label = ttk.Label(self, text="Detect",font=("Simplifica",22))
        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")
        text.pack()

```

```

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

```

```

        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, 'lxml')
            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 SUICIDAL 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=t
k.DISABLED)
l3.pack()

```

```

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

```

```

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

```

```

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

```

```

button2 = ttk.Button(self, text="About",
                      command=lambda: controller.show_frame(PageTwo))
button2.pack()

```

```

class PageTwo(ttk.Frame):

```

```

    def __init__(self, parent, controller):
        ttk.Frame.__init__(self, parent)
        label = ttk.Label(self, text="About", font=("Simplifica", 22))
        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))
        button1.pack()

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

        button2 = ttk.Button(self, text="Detect",
                              command=lambda: controller.show_frame(PageOne))
        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)

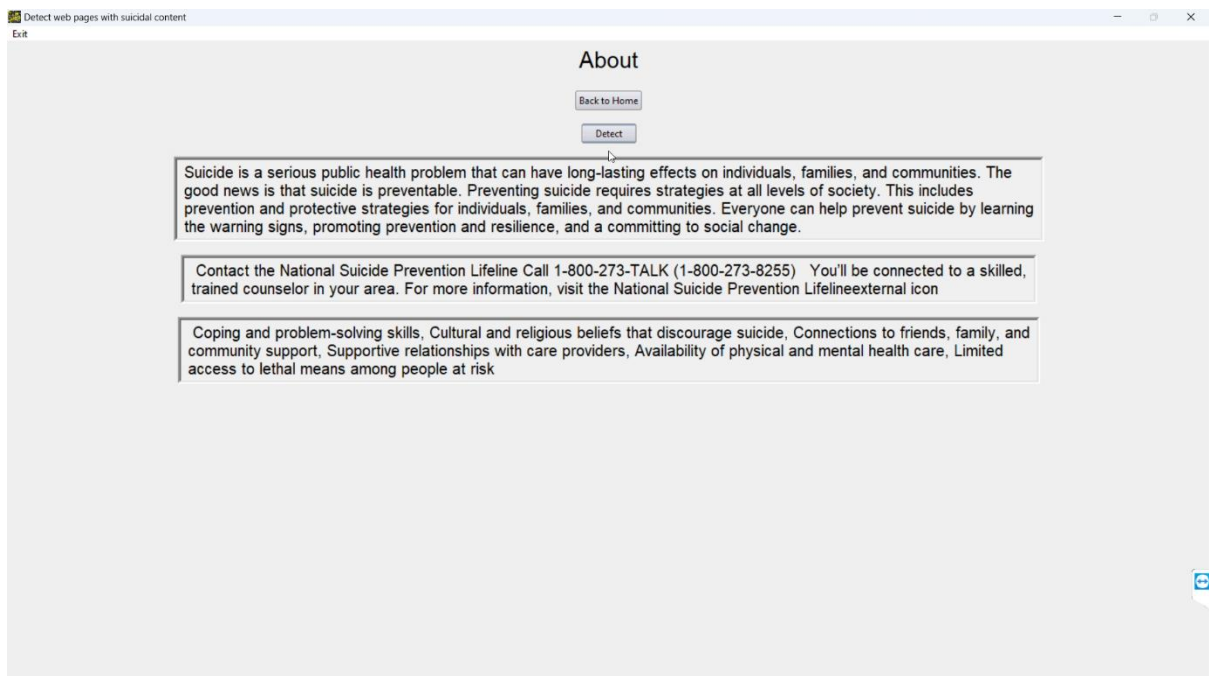
app.resizable(0,0)
app.title("Detect web pages with suicidal content")
# app title
app.state('zoomed')
app.mainloop()

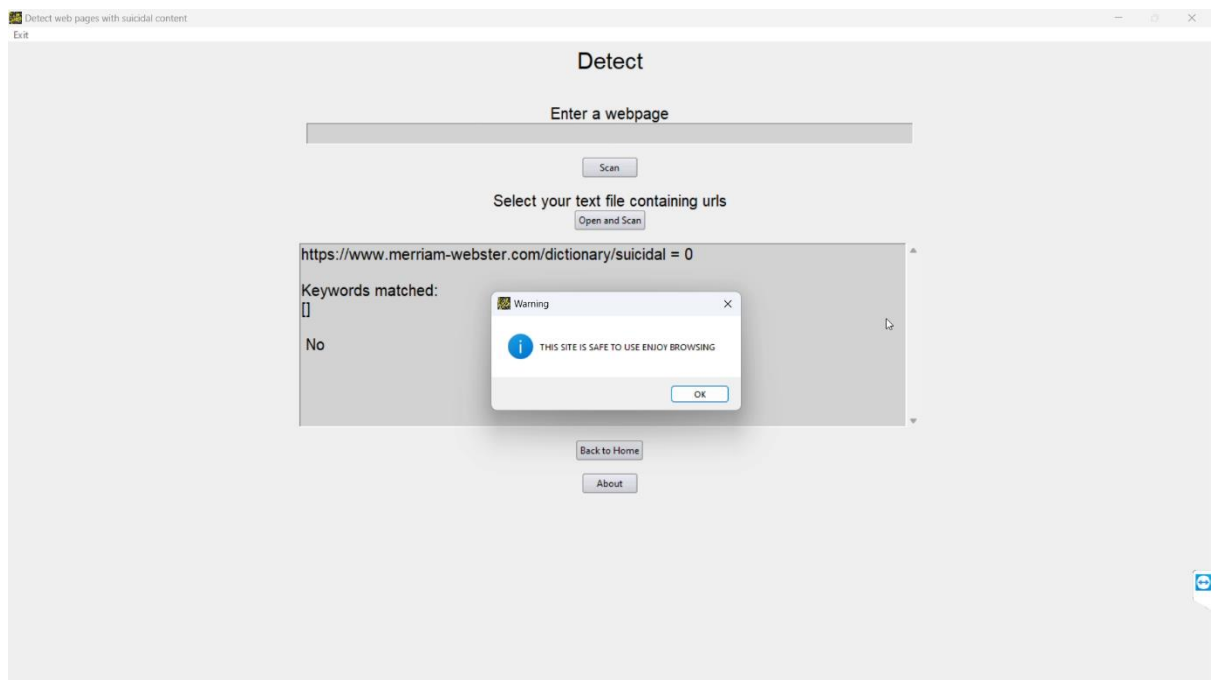
```

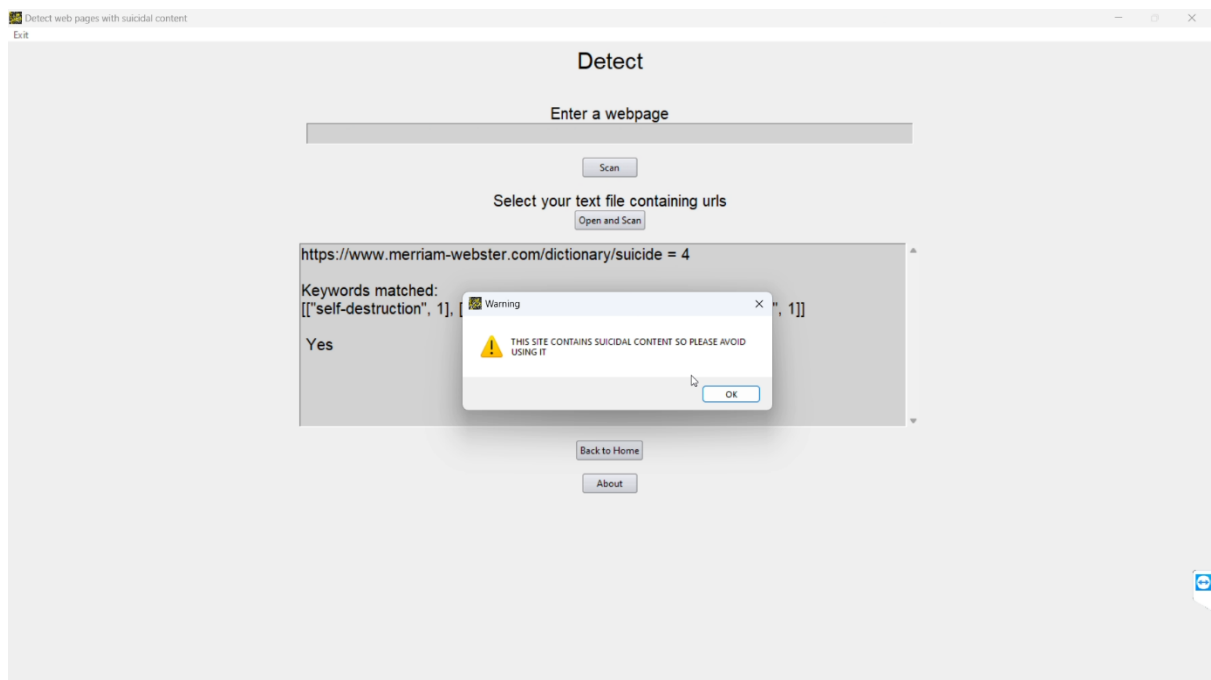
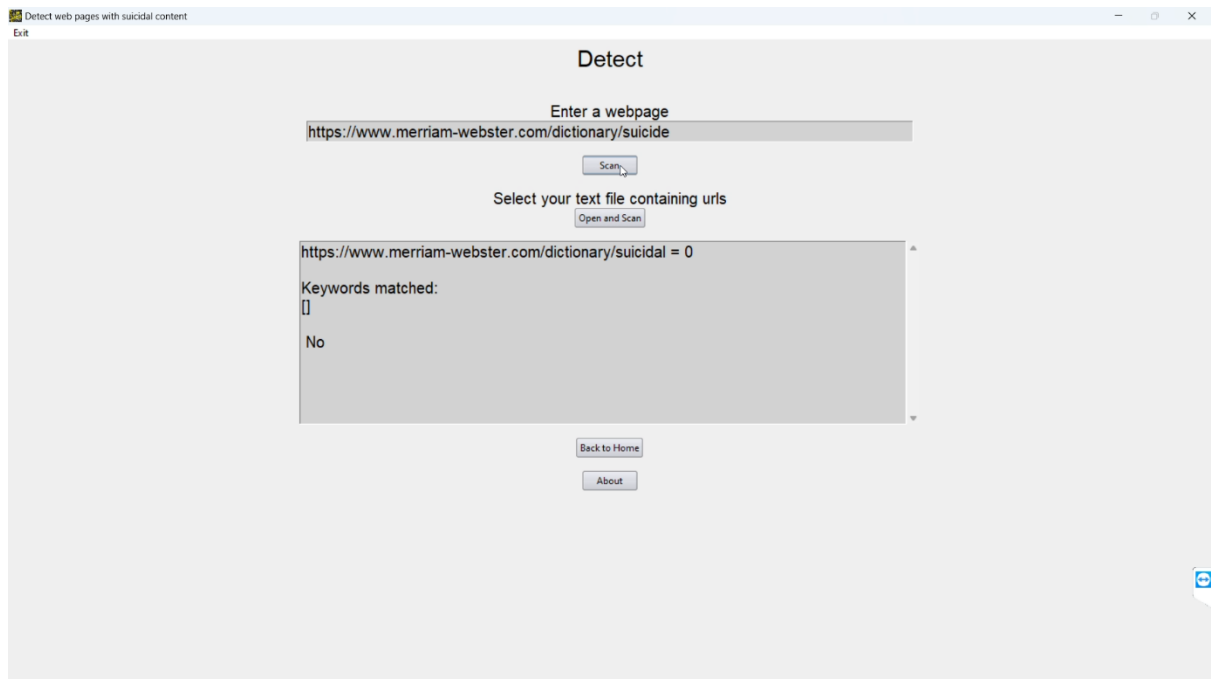
## B. SCREENSHOTS











## **A. RESEARCH PAPER**

# NLC

by venka tesh

**Submission date:** 19-Mar-2023 10:48AM (UTC-0700)

**Submission ID:** 2040678368

**File name:** Content\_Detection\_Using\_NLP\_and\_Machine\_Learning\_Technique.docx (187.64K)

**Word count:** 2076

**Character count:** 11581

## Suicidal Content Detection Using NLP and Machine Learning Technique

*Kanhai Gupta*

*Department of CSE*

*Sathyabama Institute of science and  
Technology, Chennai*

*Dr.A.C.Santha Sheela, Associate Professor*

*Department of CSE*

*Sathyabama Institute of science and  
Technology, Chennai*

# Suicidal Content Detection Using NLP and Machine Learning Technique

## ABSTRACT:

Numerous suicide studies have noted that there are about 800 000 suicides each year and that it is difficult to identify those who are suicidal. Social media usage has increased, and we have seen that users openly discuss their suicide attempts and plans on these platforms. As a result, it's critical to spot suicide risk factors early on. 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 labeled 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. The method can be used by social media platforms to automatically flag potentially concerning content for further review by trained mental health professionals who can provide support and intervention as needed. This can help prevent suicides and provide timely intervention to those who may be at risk.

**keywords** – suicidal content, webscrapping, machinelearning,beautiful soup

## INTRODUCTION

Suicide is a serious public health concern, with millions of people worldwide experiencing suicidal thoughts or attempting suicide each year. Social media platforms have become an increasingly

popular venue for people to express their struggles with mental health and suicidal ideation. Detecting suicidal content on social media can provide an opportunity to intervene and offer support to those in need. Recent advances in natural language processing (NLP) and machine learning (ML) techniques have enabled the development of algorithms that can detect suicidal content in social media data. These algorithms can analyze large volumes of social media data and identify individuals who may be at risk of suicide, allowing for timely intervention and support. Detecting suicidal content in social media data is a complex task, as it involves understanding the nuances of language and context used by individuals who express suicidal ideation. Researchers and data scientists have used a variety of NLP and ML techniques, including deep learning, transfer learning, and graph-based approaches, to develop accurate and efficient algorithms for detecting suicidal content in social media. Overall, the detection of suicidal content in social media has the potential to save lives and improve mental health outcomes. By detecting and intervening with individuals at risk of suicide, we can provide timely support and resources to those in need, ultimately reducing the incidence of suicide and improving mental health for all.

## LITERATURE REVIEW

Detecting Suicidal Ideation in Online Texts with Recurrent Neural Networks by Yashaswi Singh, Nitin Agarwal, Huan Liu (2018). In this study, the authors proposed a recurrent neural network (RNN) model for detecting suicidal ideation in online texts. They collected data from various

social media platforms and achieved promising results with an accuracy of 86.75%.

Deep learning for detecting suicidal behavior in text messages by Carlos A. Villaseñor, Eduardo F. Morales, and Rosalinda A. Vidal (2019). A deep learning model was used in this study to find suicidal behavior in text messages. The authors collected data from a crisis text line and achieved an F1 score of 0.74.

Automated Suicide Risk Assessment and Intervention Planning from Clinical Records and Natural Language Processing of Patients with Severe Mental Illnesses by Guodong Long, Benjamin Glicksberg, et al. (2020). In this study, the authors used NLP and machine learning techniques to develop an automated suicide risk assessment system. They achieved an AUC of 0.83 in predicting suicide risk.

Natural language processing and machine learning for suicide risk prediction by Wei Wu, Adarsh Vennela, et al. (2021). This study used NLP and machine learning techniques to develop a suicide risk prediction model. The authors collected data from an online mental health forum and achieved an F1 score of 0.73.

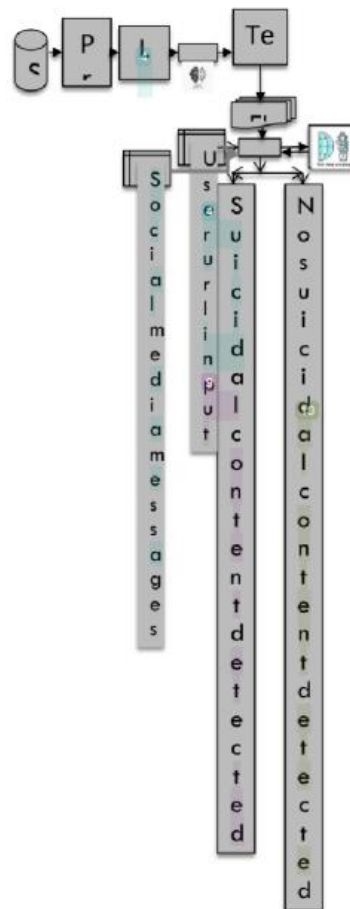
A deep learning approach for detecting suicidal ideation in social media posts by Arunima Roy, Riddhiman Dasgupta, et al. (2021). This study proposed a deep learning approach for detecting suicidal ideation in social media posts. The authors collected data from Reddit and achieved an F1 score of 0.72.

#### METHODOLOGY

The first step to gather a dataset of social media messages that are labeled as either suicidal or non-suicidal. This dataset can be created by manually reviewing a sample of messages and labeling them accordingly, or

by using a pre-existing dataset that has already been labeled. Once the dataset is prepared, the next step is to extract relevant features from the messages. This could include things like the use of certain keywords or phrases, the presence of certain emotions or sentiments, or other linguistic and contextual factors that may indicate a risk for suicide. With the features extracted, logistic regression can be used to train a model that can predict whether a given message is suicidal or not. The model is trained by adjusting the weights of the input features to maximize the likelihood of correctly classifying the training data. When attempting to predict one of two outcomes based on a collection of input features for binary classification problems, the popular machine learning technique known as logistic regression is applied. In the case of detecting suicidal content in social media messages, logistic regression can be a useful tool for identifying messages that indicate a potential risk for suicide.





After the model has been trained, it can be used to categorize new messages as suicidal or not. This can be done by feeding the input features for each new message into the trained model and observing the output prediction. Overall, logistic regression can be an effective tool for detecting suicidal content in social media messages, but it is

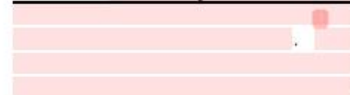
important to note that it is not a foolproof method and may produce false positives or false negatives. It is important to supplement machine learning techniques with human review and intervention to ensure the accuracy and effectiveness of suicide prevention efforts. Identify the target website(s) and determine the location of the messages you want to retrieve. This might involve examining the HTML source code of the website(s) to identify the location of the messages.

Choose a programming language and web scraping library that will allow you to extract the messages. Popular options include Python with libraries such as BeautifulSoup or Scrapy. Write a script that programmatically navigates to the target website(s) and retrieves the messages. This may involve sending requests to the website's server and parsing the HTML response to extract the desired information.

Depending on the website and the structure of the messages, you may need to use techniques such as pagination or scrolling to retrieve all of the messages. Once the messages are retrieved, you may want to clean and preprocess the data to remove any irrelevant or sensitive information. Store the retrieved messages in a suitable format, such as a text file or database, for further analysis. Meanwhile GUI is build to user interface, the same method helps to detect in social media messages also.

#### ADVANTAGES:

In order to categorize depression and its severity, machine learning techniques can be combined with NLP approaches, which concentrate on the analysis of acoustic and





task administration compared to standard assessments are all benefits of using these methods to understand depression symptoms through speech. Identification and treatment of suicidal older adults are crucial given the world's aging population. The use of NLP techniques is demonstrating promise in the assessment, monitoring, and detection of depression and other comorbidities in both younger and older people.

#### **MODULES**

- Data Collection
- Pre Processing
- Auditioning Models
- Model testing
- Web Scraping
- Real time Prediction

#### **DATA COLLECTION**

- data set--- kaggle
- data set name---reddit sentimental data

It contains a collection of Reddit posts and comments from the "SuicideWatch" subreddit, which is a community where people can share their struggles with mental health and suicidal thoughts. The dataset contains around 870000 comments posted between 2011 and 2015. The comments are labeled with tags indicating whether they contain suicidal ideation. Researchers and data scientists can use this dataset to develop and test machine learning models for detecting suicidal content in social media. The use of this dataset can aid in understanding the language and context used by individuals who express suicidal ideation, which can ultimately help in developing effective interventions and support systems for those struggling with mental health issues.

#### **PRE PROCESSING**

removeHandles(text): removes any Twitter handles (e.g., "@username") from the input text.

removeHashtag(text): removes any hashtags (e.g., "#hashtag") from the input text.

removeEmoji(text): removes any emojis from the input text.

cleanSentences(text, lemmatise=True): applies a series of text preprocessing steps (e.g., removing stopwords, stemming) to the input text, and optionally performs lemmatization. This function takes an array of strings as input and returns an array of cleaned strings.

#### **AUDITIONING MODELS LOGISTIC REGRESSION**

For binary classification problems, where the objective is to predict one of two possible outcomes based on a set of input features, logistic regression is a well-liked machine learning technique. In the case of detecting suicidal content in social media messages, logistic regression can be a useful tool for identifying messages that indicate a potential risk for suicide.

A statistical model called logistic regression is used to estimate the likelihood of a binary outcome (such as true or false, yes or no). The formula for logistic regression is:

$$p = 1 / (1 + e^{-(z)})$$

where p is the probability of the binary outcome, e is the base of the natural logarithm (approximately equal to 2.71828), and z is the log-odds or logit of the binary outcome, given by:

#### **MODEL TESTING**

**True Positive:** You predicted positive, and it's true.

**True Negative:** You predicted negative, and it's true.

**False Positive: (Type 1 Error):** You predicted positive, and it's false.

**False Negative: (Type 2 Error):** You predicted negative, and it's false.

Confusion Matrix		Target		
		Positive	Negative	
Model	Positive	a	b	Precision = $\frac{a}{a+b}$
	Negative	c	d	Recall = $\frac{a}{a+c}$
		Sensitivity = $\frac{a}{a+c}$	Specificity = $\frac{d}{b+d}$	Accuracy = $\frac{(a+d)}{(a+b+c+d)}$

- F1 SCORE
- PRECISION
- RECALL
- ACCURACY

### WEBSCRAPING

Web scraping is a technique used to extract data from websites. SCRAPING is done in project using BeautifulSoup and Requests. Use the requests library to send an HTTP request to the webpage Use BeautifulSoup to parse the HTML content of the webpage Use BeautifulSoup's functions to extract the data Process the extracted data

### OUTPUT



**Fig: Output screenshot**

The url can be pasted in first input tab and the prediction result get posted in the second tab. If any suicidal content detected it will give a pop up message

### REFERENCE PAPERS:

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. Singh, Y., Agarwal, N., & Liu, H. (2018). Detecting Suicidal Ideation in Online Texts with Recurrent Neural Networks. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (pp. 3981-3987).

7. Villaseñor, C. A., Morales, E. F., & Vidal, R. A. (2019). Deep learning for detecting suicidal behavior in text messages. In 2019 IEEE 19th International Conference on Data Mining Workshops (ICDMW) (pp. 399-406). IEEE.

8.Long, G., Glicksberg, B., Shuangshoti, S., et al. (2020). Automated Suicide Risk Assessment and Intervention Planning from Clinical Records and Natural Language Processing of Patients with Severe Mental Illnesses. *Journal of Medical Systems*, 44(7), 129.

9.Wu, W., Vennela, A., Chen, Y., et al. (2021). Natural language processing and machine learning for suicide risk prediction. In 2021 IEEE International Conference on Healthcare Informatics (ICHI) (pp. 1-8). IEEE.

10.Roy, A., Dasgupta, R., Mondal, A., et al. (2021). A deep learning approach for detecting suicidal ideation in social media posts. *IEEE Transactions on Knowledge and Data Engineering*, 33(1), 1-1.

## NLC

### ORIGINALITY REPORT

14%

SIMILARITY INDEX

11%

INTERNET SOURCES

4%

PUBLICATIONS

8%

STUDENT PAPERS

### PRIMARY SOURCES

1	<a href="http://www.ncbi.nlm.nih.gov">www.ncbi.nlm.nih.gov</a> Internet Source	3%
2	<a href="http://insightimi.wordpress.com">insightimi.wordpress.com</a> Internet Source	2%
3	Submitted to The British College Student Paper	2%
4	<a href="http://star.mpae.gwdg.de">star.mpae.gwdg.de</a> Internet Source	1%
5	Submitted to Vaasan yliopisto Student Paper	1%
6	Submitted to University of Bucharest Student Paper	1%
7	<a href="http://pure.tue.nl">pure.tue.nl</a> Internet Source	1%
8	K Lokesh, Sathyajee Srivastava, M. Praveen Kumar, S. Arun, S. Padmapriya, R. Krishnamoorthy. "Detection of Stomach Cancer Using Deep Neural Network in Healthcare Sector", 2021 3rd International	1%

Conference on Advances in Computing,  
Communication Control and Networking  
(ICAC3N), 2021

Publication

9	<a href="http://real-j.mtak.hu">real-j.mtak.hu</a> Internet Source	1 %
10	<a href="http://hdl.handle.net">hdl.handle.net</a> Internet Source	1 %
11	<a href="https://github.com">github.com</a> Internet Source	<1 %
12	Submitted to Florida International University Student Paper	<1 %
13	<a href="http://www.frontiersin.org">www.frontiersin.org</a> Internet Source	<1 %

Exclude quotes On

Exclude matches Off

Exclude bibliography On