**HEALTH DIAGNOSIS**



A project report submitted in partial fulfilment of requirements for the award of a degree of

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**by**

**BODINATHAM SAI KIRAN(219X1A05B0 )**

**RODDAM MAHMOOD SHARIFF (219X1A05G3 )**

**CHIKKALA YESHWANTH(219X1A05B5 )**

**Under the esteemed guidance of**

**M. SHASHIDHAR**

**Assistant Professor**

**Department of C.S.E.**

**Department of Computer Science and Engineering**

**G. PULLA REDDY ENGINEERING COLLEGE (Autonomous):**

**KURNOOL**

**(Affiliated to JNTUA, ANANTAPURAMU)**

**2024 – 2025**

i

**Department of Computer Science and Engineering**

**G. PULLA REDDY ENGINEERING COLLEGE (Autonomous): KURNOOL**

**(Affiliated to JNTUA, ANANTAPURAMU)**



**CERTIFICATE**

***This is to certify that the Project Work entitled*** ‘HEALTH DIAGNOSIS ’ **is a bonafide record of work carried out by**

**BODINATHAM SAI KIRAN**

**RODDAM MAHMOOD SHARIFF,**

**CHIKKALA YESHWANTH**

Under my guidance and supervision in partial fulfillment of the requirements for the award of the degree of

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

|  |  |
| --- | --- |
| **M. Shashidhar,** | **Dr. N. Kasiviswanath** |
| Assistant Professor, | Professor & Head of the Department, |
| Department of C.S.E., | Department of C.S.E., |
| G. Pulla Reddy Engineering College, | G. Pulla Reddy Engineering College, |
| Kurnool. | Kurnool. |

ii

**DECLARATION**

I hereby declare that the project titled “**HEALTH** **DIAGNOSIS** ” is an authentic work carried out by me as a student of **G. PULLA REDDY ENGINEERING COLLEGE(Autonomous) Kurnool,** during 2024-25 and has not been submitted elsewhere forthe award of any degree or diploma in part or in full to any institute.

iii

**ACKNOWLEDGEMENT**

I wish to express my deep sense of gratitude to our project guide **M.SHASHIDHAR**, Assistant Professor of Computer Science and Engineering Department, G. Pulla Reddy Engineering College, for her immaculate guidance, constant encouragement and cooperation which have made it possible to bring out this project work.

I am grateful to our project in-charge **Dr.K.ISHTHAQ AHAMED**, Associate Professor of Computer Science and Engineering Department, G. Pulla Reddy Engineering College, for helping us and giving us the required information needed for our project work.

I am thankful to our Head of the Department **Dr. N. Kasiviswanath**, for his wholehearted support and encouragement during the project sessions.

I am grateful to our respected Principal **Dr. B. Sreenivasa Reddy** for providing requisite facilities and helping us in providing such a good environment.

I wish to convey our acknowledgements to all the staff members of the Computer Science and Engineering Department for giving the required information needed for our project work.

Finally, I wish to thank all our friends and well-wishers who have helped us directly or indirectly during the course of this project work.

iv

**ABSTRACT**

The core of the project involves creating a suite of machine learning algorithms designed to analyze a range of diagnostic data, including patient symptoms, clinical histories, lab results, and imaging studies. By employing techniques such as supervised learning, neural networks, and ensemble methods, the project seeks to build models that can predict the likelihood of various conditions based on historical and real-time data.

A project would be very useful in the medical field and machine learning- based web application would be created for medical diagnosis. For a medical diagnosis, a machine learning model would be developed and integrated with the created web application. The user would be able to upload his medical data on the web application. The web application would pass this data to a developed machine learning model for health

disease detection. After analysing the user’s data it predicts the health condition of the user

**TABLE OF CONTENTS**

[1. INTRODUCTION](#page9) [1](#page9)

[1.1 INTRODUCTION](#page9) [1](#page9)

[1.2 MOTIVATION](#page10) [2](#page10)

[1.3 PROBLEM DEFINITION](#page11) [3](#page11)

[1.4 OBJECTIVES OF THE PROJECT](#page11) [3](#page11)

[1.5 LIMITATIONS OF THE PROJECT](#page13) [5](#page13)

[1.6 ORGANIZATION OF THE REPORT](#page14) [6](#page14)

[2. SYSTEM SPECIFICATIONS](#page16) [7](#page16)

[2.1 SOFTWARE SPECIFICATIONS](#page16) [7](#page16)

[2.2 HARDWARE SPECIFICATIONS](#page17) [8](#page17)

[3. LITERATURE SURVEY](#page19) [9](#page19)

[3.1 INTRODUCTION](#page19) [9](#page19)

[3.2 EXISTING SYSTEM](#page20) [10](#page20)

[3.3 LIMITATIONS OF EXISTING SYSTEM](#page22) [12](#page22)

[3.4 PROPOSED SYSTEM](#page23) [13](#page23)

[4. DESIGN AND IMPLEMENTATION](#page27) [16](#page27)

[4.1 INTRODUCTION](#page27) [16](#page27)

[4.2 UML DIAGRAMS](#page27) [16](#page27)

[4.3. IMPLEMENTATION](#page33) [22](#page33)

[4.4 SOURCE CODE](#page38) [27](#page38)

[4.5 OUTPUTS](#page54) [43](#page54)

[5. CONCLUSION](#page58) [46](#page58)

[5.1 CONCLUSION](#page58) [46](#page58)

[5.2 FUTURE ENHANCEMENT](#page58) [46](#page58)

[REFERENCES](#page60) [48](#page60)

vi

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **FIGURE NAME** | **PAGE NO.** |
| 4.2.1 | Use Case Diagram | 18 |
| 4.2.2 | Sequence Diagram | 19 |
| 4.2.3 | Class Diagram | 20 |
| 4.2.4 | Activity Diagram | 21 |
| 4.3.1 | Methods and Categorization | 22 |
| 4.3.5 | Project Flow Chart | 26 |
| 4.4.1 | Data Visualization | 30 |
| 4.5.1 | Web Interface | 43 |
| 4.5.2 | Input - Suicidal Text | 44 |
| 4.5.3 | Suicidal Class Prediction | 44 |
| 4.5.4 | Input – Non-Suicidal Text | 45 |
| 4.5.5 | Non-Suicidal Class Prediction | 45 |

vii

HEALTH DIAGNOSIS



**INTRODUCTION**



DEPT OF CSE, GPREC, KNL

HEALTH DIAGNOSIS



**1. INTRODUCTION**

**1.1 INTRODUCTION**

In today’s rapidly evolving world, healthcare remains a fundamental pillar of human well-being, with the early diagnosis and prevention of diseases playing a vital role in saving lives and improving quality of life. Among the various health conditions that pose a significant threat to global populations, chronic illnesses such as heart disease, diabetes, and kidney disease continue to dominate as leading causes of death and long-term disability. These diseases often progress silently, showing symptoms only in advanced stages, which makes early detection critically important. Traditional diagnostic methods, while accurate, are often limited by factors such as time consumption, high costs, dependency on medical infrastructure, and the need for expert interpretation. These challenges are especially pronounced in rural and underserved regions, where access to healthcare facilities and skilled professionals is minimal. In recent years, the emergence of advanced technologies, particularly machine learning (ML), has opened new frontiers in medical diagnostics by enabling the development of predictive models that can analyze complex medical data and detect diseases at an early stage. Machine learning algorithms have the capability to process large datasets consisting of patient health records, laboratory results, lifestyle factors, and clinical parameters to identify hidden patterns that may not be immediately apparent through traditional analysis. This project aims to harness the power of machine learning to build predictive models that can accurately forecast the likelihood of heart disease, diabetes, and kidney disease in individuals based on relevant health indicators such as age, gender, blood pressure, glucose levels, BMI, cholesterol, creatinine levels, and other biomarkers. These models not only provide rapid, cost-effective, and scalable diagnostic support to medical professionals but also empower individuals to monitor their health and take proactive steps toward prevention. Furthermore, integrating these models into user-friendly platforms like mobile apps or web interfaces can democratize access to healthcare insights, allowing users to perform self-assessments and seek timely medical intervention. By leveraging intelligent algorithms, this approach aligns with the vision of preventive healthcare, reduces the burden on healthcare systems, and contributes to building a data-driven, digitally empowered healthcare ecosystem where diseases are identified and addressed before they escalate into critical conditions. Furthermore, with the increasing digitization of healthcare records and the availability of large-scale medical datasets, machine learning provides a scalable solution to mine actionable insights that were previously inaccessible through conventional means. These intelligent systems not only reduce the diagnostic burden on healthcare professionals but also minimize human error, thus ensuring more consistent and objective evaluation of patient health. As wearable devices and IoT-enabled health monitors become more widespread, real-time data collection integrated with machine learning algorithms can enable continuous health surveillance and early warnings for high-risk individuals. By incorporating such technologies, the healthcare industry can transition from reactive treatment models to proactive, preventive care systems that focus on maintaining long-term wellness rather than just addressing illness



DEPT OF CSE, GPREC, KNL **1** | P a g e

HEALTH DIAGNOSIS



classification is to determine whether candidates, through their posts, have suicidal ideations.

Machine learning methods and NLP also will be applied in this field.

**1.2 MOTIVATION**

The increasing prevalence of chronic diseases such as heart disease, diabetes, and kidney disease presents a growing challenge to healthcare systems worldwide. Millions of people are affected by these conditions each year, leading to significant health complications, reduced quality of life, and an enormous strain on healthcare resources. Early detection and intervention are critical in managing these diseases effectively, but traditional diagnostic methods can be time-consuming, costly, and inaccessible to many individuals, especially in underserved regions. With the advent of machine learning and its ability to process vast datasets and recognize complex patterns, there is an unparalleled opportunity to revolutionize the way chronic diseases are diagnosed and managed. By developing predictive models that can analyze patient data and detect early warning signs, this project aims to empower healthcare providers to make faster, more accurate decisions and enable individuals to take proactive steps toward maintaining their health. The motivation behind this work is to leverage cutting-edge technology to improve patient outcomes, reduce the burden of preventable diseases, and make healthcare more efficient, scalable, and accessible to all.

**1.Early Intervention**

The primary purpose is to detect signs of Kidney disease ,heart attacks and diabetes health Issues. Machine learning models can analyze various data sources, such as text messages, social media posts to identify patterns indicative of distress.

**2.** **Faster and Efficient Risk Identification**

Machine learning algorithms can process massive amounts of health-related data swiftly and accurately, allowing for the early identification of high-risk cases. This reduces the time taken for diagnosis and enables quicker medical intervention, which can significantly improve patient outcomes.

**3. Large-Scale Disease Screening**

Automated diagnosis systems offer high scalability, making them suitable for screening large populations in hospitals, clinics, or even through health apps. This ensures wider reach, including in rural or low-resource settings where regular check-ups might be limited.

**4. Personalized Diagnostic Insights**

Machine learning models can be customized to consider individual-specific health attributes such as age, gender, genetic predisposition, and lifestyle habits and get the health predictions.



DEPT OF CSE, GPREC, KNL **2** | P a g e

HEALTH DIAGNOSIS



**5. Proactive Health Monitoring**

The foremost objective is to recognize early indicators of chronic diseases before they escalate. Machine learning models can Analyze medical data such as patient history, clinical test results, and lifestyle factors to detect subtle patterns and warning signs that might go unnoticed through traditional diagnosis..

**6.Resource Allocation**

By identifying and prioritizing individuals at higher risk, limited mental health resources can be allocated more efficiently. This helps ensure that those who need support the most receive it in a timely manner.

It's important to approach such projects with ethical considerations, including user consent, data privacy, and the potential for unintended consequences. Collaboration with mental health professionals, ethicists, and affected communities is crucial in developing responsible and effective solutions.

**1.3 PROBLEM DEFINITION**

The problem definition for a health diagnosis system using machine learning involves clearly outlining the objectives, scope, and expected outcomes of the project. In this context, the task is to design and implement a predictive model that can accurately identify individuals at risk of chronic health conditions such as heart disease, diabetes, and kidney disease. The goal is to enable early detection and proactive medical intervention by analyzing patient data such as age, blood pressure, glucose levels, cholesterol, body mass index, and other relevant clinical attributes. The machine learning model should be capable of classifying health conditions with a high degree of accuracy, sensitivity, and reliability. It must also be scalable and adaptable to different population groups and healthcare settings. Moreover, the system should act as a decision-support tool for medical professionals and offer valuable insights for patients, empowering them to take preventive measures. The solution must be ethically designed, ensuring data privacy, fairness, and interpretability while aiming to reduce the burden of these life-threatening diseases through timely and intelligent diagnosis.

**1.4 OBJECTIVES OF THE PROJECT**

The objectives of a Health diagnosis machine learning project are multifaceted and centered around leveraging technology to identify and address the risk of heart attacks ,kidney and diabetes . Here are key objectives for such a project:







DEPT OF CSE, GPREC, KNL **3** | P a g e

HEALTH DIAGNOSIS



**1. Early Detection**

Develop a machine learning model capable of early detection of disease by analyzing textual data from various sources like parameters. The goal is to identify disease scenario.

**2. Scalability**

Design the system to be scalable, allowing for the analysis of large volumes of data efficiently. This scalability is essential for monitoring diverse online platforms where individuals may express their thoughts and emotions.

**3. User Privacy Protection**

Prioritize the protection of user privacy by implementing mechanisms to anonymize and secure sensitive information. Build trust with users by ensuring that their personal data is handled responsibly.

**5. Integration with Support Services**

Establish seamless integration with mental health support services and professionals. Provide a mechanism for automatically alerting relevant stakeholders when the model identifies high-risk cases, ensuring that individuals receive timely assistance.

**6. Ethical Guidelines**

Adhere to ethical guidelines in the development and deployment of the model. Prioritize transparency, accountability, and the responsible use of technology to address mental health challenges.

**7. User Awareness and Education**

Develop strategies to raise user awareness about the purpose and capabilities of the system. Educate users about the importance of seeking professional help and the limitations of automated detection systems.

**8. Model Accuracy and Performance Optimization**

Continuously improve the performance of the machine learning model by fine-tuning hyperparameters, selecting optimal features, and applying suitable algorithms. The objective is to achieve high accuracy, precision, recall, and F1-score in predicting disease outcomes, ensuring the system is reliable for real-world healthcare applications





DEPT OF CSE, GPREC, KNL **4** | P a g e

HEALTH DIAGNOSIS



**1.5 LIMITATIONS OF THE PROJECT**

While suicidal ideation detection machine learning projects hold promise for early intervention and support, it's crucial to be aware of their limitations. Here are some common limitations associated with such projects:

**1.False Positives and False Negatives**

The model may produce false positives (predicting a disease risk when it is not present) or false negatives (failing to identify a true risk). These misclassifications can lead to unnecessary anxiety, delayed treatment, or missed medical intervention, which may affect patient outcomes.

**2**. **Limited Contextual Insight**

Machine learning models rely heavily on structured data and may not fully understand the broader medical context behind certain symptoms or parameters. Factors like environmental influences, family history, and patient lifestyle may not be captured comprehensively, impacting the accuracy of the diagnosis.

**3. Data Quality and Availability**

The performance of the model is highly dependent on the quality, completeness, and consistency of the input data. Missing or inaccurate medical records can reduce the reliability of predictions and lead to biased or skewed results.

**4. Generalization Across Populations**

Models trained on specific datasets may not generalize well across different demographics or populations. Variations in genetic makeup, regional healthcare standards, and lifestyle patterns can limit the model's effectiveness when applied to a diverse user base.

5. **Evolution of Medical Standards**

Medical knowledge and diagnostic standards evolve over time. A model trained on outdated clinical data might become less effective or even irrelevant unless regularly updated to reflect current medical guidelines and practices.

**6. Ethical and Privacy Concerns**

Collecting and analyzing sensitive health data raises significant privacy concerns. Ensuring data anonymization, secure storage, and user consent is essential. Ethical considerations must also be addressed, particularly regarding the potential misuse of health predictions and decisions made solely by automated systems.







DEPT OF CSE, GPREC, KNL **5** | P a g e

HEALTH DIAGNOSIS



**1.6 ORGANIZATION OF THE REPORT**

This is to follow up the next chapters i.e., Chapter 2 contains the information about the system specifications. It clearly explains the tools and frameworks of the project and their usage. Software requirements and hardware requirements are also mentioned in the chapter. The next chapter i.e., Chapter 3 deals with the Literature Survey of the project. It covers the papers published on the Health Diagnosis. The next chapter i.e., Chapter 4 deals with the design and implementation of the project. It covers the technology that is used for the project i.e., it also contains the source code of the project and the output screenshots of the project. The last chapter i.e., Chapter 5 provides the concluding information of the project. The report ends with a list of references that have been used.



DEPT OF CSE, GPREC, KNL **6** | P a g e

HEALTH DIAGNOSIS



**SYSTEM SPECIFICATIONS**



DEPT OF CSE, GPREC, KNL

HEALTH DIAGNOSIS



**2. SYSTEM SPECIFICATIONS**

There are the requirements for doing the project. Without using these tools and software, we cannot do the project. So, we need two requirements to do the project. They are

1. Hardware Requirements.
2. Software Requirements.

**2.1 SOFTWARE SPECIFICATIONS**

Software specifications for Health Diagnosis are

1. Development Environment:
   * The development environment for the Health Diagnosis Project includes Google Collab for writing and executing Python code through a browser, leveraging machine learning.
   * Additionally, Visual Studio Code (VS Code) serves as the preferred editor for building optimized and refined web applications.
2. Backend Technologies:
   * Python programming language for backend development.
   * Machine learning technique for predictive analysis.
3. Frontend Technologies:
   * For the frontend, Python is used, ensuring a visually appealing and responsive design.
   * CSS frameworks like Bootstrap enhance the web application's responsiveness.
4. Testing Frameworks:
   * Testing processes are conducted using Google Colab and Visual Studio Code, ensuring rigorous evaluation of the system.
5. Deployment and hosting:
   * Integration of the frontend with the backend is achieved through Flask and Streamlit application deployment, allowing the machine learning model to be hosted on a web browser.
6. Additional libraries and datasets:
   * Libraries such as numpy, pandas, scikit learn to enhance the performance of the model.



DEPT OF CSE, GPREC, KNL **7** | P a g e

HEALTH DIAGNOSIS



* Dataset is a CSV file consisting of two columns: one text column, one target variable which has two classes (Binary values - 0 or 1).

**2.2 HARDWARE SPECIFICATIONS**

Hardware specifications for Health Diagnosis are

1. **System**
   * The system has to be equipped with a Pentium i3 or i5 processor (which is a basic requirement; advanced processors are supported too), ensuring efficient processing capabilities to support the functionality.
2. **Hard Disk**
   * A 500 GB hard disk is included, providing ample storage space for datasets, models, and other necessary components of the project.
3. **Display**
   * Featuring a 15-inch LED monitor, the system ensures a clear and visually accessible interface for users and developers involved in the project.
4. **Input Devices**
   * Standard input devices, including a keyboard and mouse, are incorporated, facilitating user interaction and system operation.
5. **Random Access Memory (RAM)**
   * With 4 GB of RAM, the system possesses the memory capacity necessary for smooth execution of processes and applications associated with the project.



DEPT OF CSE, GPREC, KNL **8** | P a g e

HEALTH DIAGNOSIS



**LITERATURE SURVEY**



DEPT OF CSE, GPREC, KNL

HEALTH DIAGNOSIS



**3. LITERATURE SURVEY**

**3.1 INTRODUCTION**

In medical domains, artificial intelligence (AI) primarily focuses on developing the algorithms and techniques to determine whether a system’s behavior is correct in disease diagnosis. Medical diagnosis identifies the disease or conditions that explain a person’s symptoms and signs. Typically, diagnostic information is gathered from the patient’s history and physical examination. It is frequently difficult due to the fact that many indications and symptoms are ambiguous and can only be diagnosed by trained health experts. Therefore, countries that lack enough health professionals for their populations, such as developing countries like Bangladesh and India, face difficulty providing proper diagnostic procedures for their maximum population of patients. Moreover, diagnosis procedures often require medical tests, which low-income people often find expensive and difficult to afford.

As humans are prone to error, it is not surprising that a patient may have overdiagnosis occur more often. If overdiagnosis, problems such as unnecessary treatment will arise, impacting individuals’ health and economy. According to the National Academics of Science, Engineering, and Medicine report of 2015, the majority of people will encounter at least one diagnostic mistake during their lifespan. Various factors may influence the misdiagnosis, which includes:

* lack of proper symptoms, which often unnoticeable
* the condition of rare disease
* the disease is omitted mistakenly from the consideration

Machine learning (ML) is used practically everywhere, from cutting-edge technology (such as mobile phones, computers, and robotics) to health care (i.e., disease diagnosis, safety). ML is gaining popularity in various fields, including disease diagnosis in health care. Many researchers and practitioners illustrate the promise of machine-learning-based disease diagnosis (MLBDD), which is inexpensive and time-efficient. Traditional diagnosis processes are costly, time-consuming, and often require human intervention. While the individual’s ability restricts traditional diagnosis techniques, ML-based systems have no such limitations, and machines do not get exhausted as humans do. As a result, a method to diagnose disease with outnumbered patients’ unexpected presence in health care may be developed.

Health diagnosis using machine learning (ML) has gained significant attention due to its ability to analyze vast amounts of medical data and identify patterns that humans might overlook. Diseases such as heart disease, kidney disease, and diabetes can be predicted using ML algorithms trained on patient records, including symptoms, medical history, and test results. The integration of AI-driven healthcare systems can assist in early diagnosis, thereby improving treatment outcomes and reducing mortality rates.

Recent breakthroughs in ML difficulties, such as imbalanced data, ML interpretation, and ML ethics in medical domains, are only a few of the many challenging fields to handle in a nutshell. In this paper, we provide a review that highlights the novel uses of ML and DL in disease diagnosis and gives an overview of development in this field in order to shed some light on this current trend, approaches, and issues connected with ML in disease diagnosis. We begin by outlining several methods to machine learning and deep learning techniques and particular architecture for detecting and categorizing various forms of disease diagnosis.

Machine learning (ML) is an approach that analyzes data samples to create main conclusions using mathematical and statistical approaches, allowing machines to learn without programming. Arthur Samuel presented machine learning in games and pattern recognition algorithms to learn from experience in 1959, which was the first time the important advancement was recognized. The core principle of ML is to learn from data in order to forecast or make decisions depending on the assigned task. Thanks to machine learning (ML) technology, many time-consuming jobs may now be completed swiftly and with minimal effort. With the exponential expansion of computer power and data capacity, it is becoming simpler to train data-driven ML models to predict outcomes with near-perfect accuracy. Several papers offer various sorts of ML approaches

The ML algorithms are generally classified into three categories such as supervised, unsupervised, and semisupervised. However, ML algorithms can be divided into several subgroups based on different learning approaches. Some of the popular ML algorithms include linear regression, logistic regression, support vector machines (SVM), random forest (RF), and naïve Bayes (NB)

Medical diagnostics can use ML for various tasks, including medical image analysis, health care analysis, disease detection, risk assessment, and personalized treatment recommendations. Medical diagnosis in machine learning processes involves the utilization of advanced algorithms to analyze patient data and provide accurate insights for healthcare professionals.



DEPT OF CSE, GPREC, KNL **9** | P a g e

SUICIDAL THOUGHT DETECTION



Additionally, AI and machine learning in healthcare industry can analyze X-rays and MRI scans, examine patient information, including test results and medical history, and offer customized treatment plans.

**3.2 EXISTING SYSTEM**

Many preliminary works have been conducted for suicidal ideation detection, especially boosted by manual feature engineering and DNN-based representation learning techniques. However, current research has several limitations, and there are still great challenges for future work.

Different machine learning techniques have been used, studied and gauged their effectiveness for suicidal tendency detection to prove that Machine Learning Algorithms like Random Forest, Naive Bayes can correctly identify residing Suicidal Tendency of a Social Media user.

1. **Suicidal Tendency Detection**
   * This paper was written by the Department of Information and Communication Technologies, University at Pompeu Fabra, Barcelona, Spain and it was published by IEEE in the year 2019.
   * They have used the Convolutional Neural Network (CNN) in this project to get an accuracy of 77%. The drawback of this approach was that they have used image based predictive models. For detecting depression and suicide.
2. **Performance Evaluation of Different Machine Learning Techniques using Twitter Data for Identification of Suicidal Intent**
   * This paper was written by Anirudh Ramachandran, Akshara Gadwe, Dishank Poddar, Saurabh Satavalekar and Sunita Sahu and was published by IEEE in 2020 at the International Conference on Electronics and Sustainable Communication Systems (ICESC).
   * Research and Evaluation based on online behavior have been conducted repeatedly. Using machine learning, this online trail of data that a person leaves behind can be used to gain insights on the behavior and psychological status.
   * In this paper, different machine learning techniques have been used, studied and gauged their effectiveness for suicidal tendency detection to prove that Machine Learning



DEPT OF CSE, GPREC, KNL **10** | P a g e

SUICIDAL THOUGHT DETECTION



algorithms like Logistic Regression can correctly identify the residing Suicidal Tendency of a Twitter user.

* + They have used algorithms such as Linear Regression, Logistic Regression, Naive Bayes, Random Forest, GBDT, XGBoost, MLFFNN on the Twitter dataset to achieve an accuracy of 76.3%. The only drawback of this approach is that it uses logistic regression which is a long process for the detection of suicides.

1. **Depression and Suicidal Tendency Identification**
   * This paper was written by Seung Young Ryu, Hyeongrae Lee, Dong-Kyun Lee AND Kyeongwoon Park Department of Mental Health Research, National Center for Mental Health, Seoul, Republic of Korea and was published on 30th December 2019.
   * This approach uses the Random Forest algorithm to get an accuracy of 85%.
2. **Prediction of Suicide Cases in India Using Machine Learning**
   * This study is used to analyze the pattern of the registered suicide cases in India. After the analysis of the available data, we can conclude that ratio of the suicide cases for men is comparatively higher than the women.
   * Also, we have identified that the most of the men who attempt or commit suicide belongs to the age group of 30 to 44 whereas the most of the ladies who commit or attempt suicide belongs to the age bracket of 15 to 29.
   * For this research we have developed the two models of Machine learning which are the neural network and SVM for estimating the causes of suicides in future to analyze the accuracy of the both models.
   * We have learned that for this kind of dataset, the Neural networks gives 77.5 % accurate results for the estimation which leads to the 17% of incorrect predictions whereas the SVM model gives a prediction accuracy of 81.5% for predicting the causes of suicide which makes SVM slightly better than neural network for this particular research.
   * In the future, the applied model of this research can be used to predict the causes independently for every age group and also to classify the causes according to the male and female separately. Also, this research could have been utilized to predict the amount of suicide promptly.



DEPT OF CSE, GPREC, KNL **11** | P a g e

SUICIDAL THOUGHT DETECTION



1. **Detection of Suicidal Ideation on Social-media: Multimodal, Relational, and Behavioral Analysis**
   * This paper was written by Diana Ramírez-Cifuentes, Ana Freire, Ricardo Baeza-Yates, Joaquim Puntí, Pilar Medina-Bravo, Diego Alejandro Velazquez, Josep Maria Gonfau, Jordi Gonzàlez and was published in the year 2020 in the Journal of Medical Internet Research.
   * This paper aimed to describe an approach for the suicide risk assessment of Spanish-speaking users on social media.
   * We aimed to explore behavioral, relational, and multimodal data extracted from multiple social platforms and develop machine learning models to detect users at risk.
   * They characterized users based on their writings, posting patterns, relations with other users, and images posted.
   * They also evaluated statistical and deep learning approaches to handle multimodal data for the detection of users with signs of suicidal ideation. To evaluate the performance of the models, they distinguished 2 control groups:
     1. users who make use of suicide-related vocabulary (focused control group)
     2. generic random users (generic control group)
   * The algorithms used were random forest, multilayer perceptron, logistic regression, and support vector machines as classifiers which gave an accuracy of 82%. But this is just an observational study.
   * Results can be improved by enhancing the contribution of the textual and relational features.

**3.3 LIMITATIONS OF EXISTING SYSTEM**

**a) Human error & misdiagnosis**

Human error and misdiagnosis occur when physicians misinterpret symptoms, leading to incorrect treatment decisions. This can happen due to factors like similar symptoms across diseases, time constraints, fatigue, or incomplete patient history. For example, early signs of heart disease, kidney disease, or diabetes may resemble less serious conditions, causing delays in accurate diagnosis.



DEPT OF CSE, GPREC, KNL **12** | P a g e

SUICIDAL THOUGHT DETECTION



**b) Late detection of diseases**

Late detection of diseases occurs when symptoms manifest only in advanced stages, making early diagnosis and treatment challenging. This is common in conditions like heart disease, kidney disease, and diabetes, where initial signs are mild or silent. By the time noticeable symptoms appear, the disease may have progressed significantly, leading to complications such as organ failure, irreversible damage, or life-threatening events like heart attacks or strokes.

**c) Time & cost factors**

Extensive medical evaluations and diagnostic tests are often necessary to confirm diseases like heart disease, kidney disease, and diabetes, but they can be time-consuming and expensive. Patients may require multiple visits, lab tests, imaging scans, and specialist consultations, which increase healthcare costs. Additionally, delays in test results may slow down diagnosis and treatment, potentially worsening the condition.

**d) Lack of predictive capability**

Traditional healthcare systems primarily rely on reactive diagnosis, meaning they detect diseases based on symptoms after they appear, rather than predicting risks early. Since they do not analyze historical patient data or emerging health trends, they miss opportunities for preventive care.

**3.4 PROPOSED SYSTEM**

The methodology is to implement a machine learning classifier to improve the performance of a language modeling and text classification for detecting heart, kidney and diabetes diseases. The algorithms that are used are:

**1. Support Vector Machine Algorithm**

Support Vector Machine or SVM can be used for both classification and regression. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. The training set will be fitted to the SVM classifier. To create the SVM classifier, we will import the SVC class from Sklearn.svm library.



DEPT OF CSE, GPREC, KNL **13** | P a g e

SUICIDAL THOUGHT DETECTION



**2. Naive Bayes Algorithm**

Naive Bayes classifiers are a collection of classification algorithms based on Bayes’ Theorem. Bayes’ Theorem finds the probability of an event occurring given the probability of another event that has already occurred. Bayes’ theorem is stated mathematically as the following equation:

P(A|B) = [P(B|A) P(A)]/P(B)

where A and B are events and P(B) ≠ 0.

The naïve Bayes (NB) classifier is a Bayesian-based probabilistic classifier. Based on a given record or data point, it forecasts membership probability for each class. The most probable class is the one having the greatest probability. Instead of predictions, the NB classifier is used to project likelihood

**3. Decision Tree Algorithm**

Decision tree is the most powerful and popular tool for classification and prediction. A decision tree is a flowchart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

**4. Logistic Regression**

Logistic regression (LR) is an ML approach that is used to solve classification issues. The LR model has a probabilistic framework, with projected values ranging from 0 to 1. Examples of LR-based ML include spam email identification, online fraud transaction detection, and malignant tumor detection. The cost function, often known as the sigmoid function, is used by LR. The sigmoid function transforms every real number between 0 and 1.

**5. Random Forest Classifier**

The Random Forest Classifier is an ensemble learning method that constructs multiple decision trees during training, offering robust and accurate predictions by aggregating the results of individual trees. It mitigates overfitting, enhances generalization, and provides insight into feature importance. In the context of suicide ideation detection, Random Forest proves versatile



DEPT OF CSE, GPREC, KNL **14** | P a g e

SUICIDAL THOUGHT DETECTION



for analyzing text-based features, demonstrating its effectiveness in capturing complex relationships within the data.

Among all the above models Support Vector Machine and Logistic Regression works well giving an accuracy of 91.6% and 91.3% respectively.



DEPT OF CSE, GPREC, KNL **15** | P a g e

SUICIDAL THOUGHT DETECTION



**DESIGN AND IMPLEMENTATION**



DEPT OF CSE, GPREC, KNL

HEALTH DIAGNOSIS



**4. DESIGN AND IMPLEMENTATION**

**4.1 INTRODUCTION**

Suicidal Thought Detection, built using machine learning model facilitates the detection of a suicidal thought in texts. It effectively predicts whether the text entered by end user falls under suicidal or non-suicidal class. Through this, we can detect if a person’s thoughts are much focused on suicide; by which we can prevent the case of suicide happening in the future. The design methodology describes the UML diagrams, implementation of the project, source code, and the output.

**4.1.1 PURPOSE**

The aim of the project is to develop an automated Suicidal Thought Detection System using machine learning and NLP. Focused on social media posts, the system seeks to proactively identify and classify signs of suicidal ideation. The main purpose of the project is to enable earlier detection, complement traditional prevention methods, and contribute to suicide prevention efforts. It helps in providing technological support to the Nongovernmental organizations in order to help the people who are in need of assistance. Ethical considerations, including privacy and consent, are prioritized in the design, and successful implementation is expected to positively impact mental health interventions and research in suicide prevention.

**4.1.2 SCOPE**

The scope of the project is the Earlier Detection of Suicidal Ideation (DSI). This involves developing a classification model capable of categorizing social media posts into classes that determine whether the user exhibits suicidal tendencies. The focus is on leveraging technology to intervene early and prevent potential fatalities.

**4.2 UML DIAGRAMS**

Unified Modeling Language (UML) diagrams are standardized visual representations used in software engineering. They include Use Case Diagrams for interactions, Class Diagrams for structure, Sequence Diagrams for chronological interactions, Activity Diagrams for workflows,



DEPT OF CSE, GPREC, KNL **16** | P a g e

HEALTH DIAGNOSIS



and State Machine Diagrams for state transitions. UML diagrams enhance communication, simplify system comprehension, and serve as blueprints for software development by providing a standardized visual language.

**4.2.1 USE CASE DIAGRAM**

The Use Case Diagram portrays the fundamental user-system interactions in the suicide detection system. It encapsulates key actions: text entry, analysis initiation, class prediction request, and the system's delivery of class and probability information. This diagram succinctly illustrates the essential functionalities triggered by the user and facilitated by the system for analyzing potentially suicidal text.



DEPT OF CSE, GPREC, KNL **17** | P a g e

SUICIDAL THOUGHT DETECTION

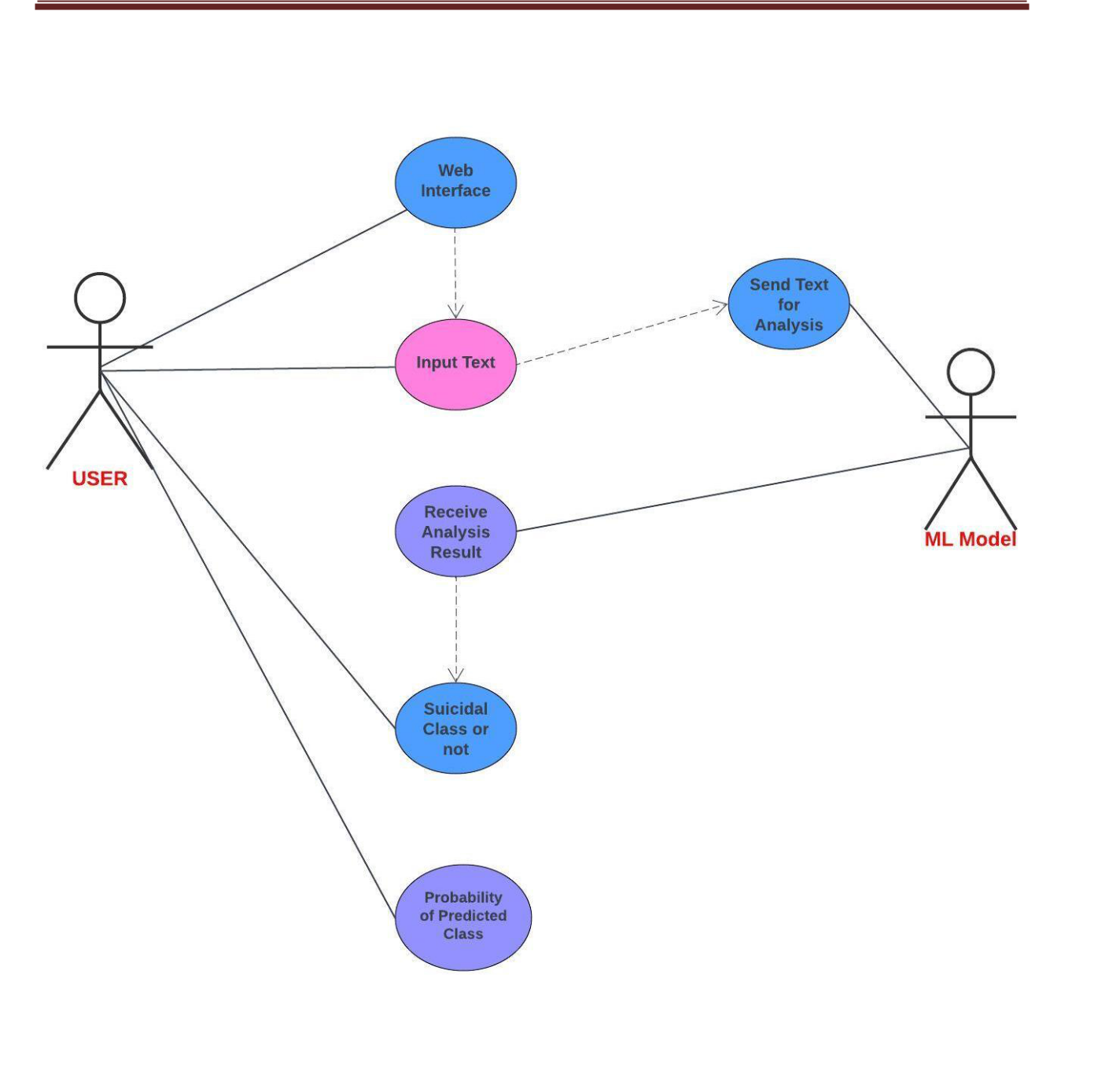


Fig 4.2.1 Use Case Diagram

**4.2.2 SEQUENCE DIAGRAM**

The Sequence Diagram visually outlines the suicide detection system's interaction flow. It depicts user input initiation, system analysis, and communication with the ML Model, summarizing the sequential process. This diagram encapsulates the system's steps, highlighting user-triggered analysis and prediction, concluding with the type of class the text belongs to.



DEPT OF CSE, GPREC, KNL **18** | P a g e

SUICIDAL THOUGHT DETECTION

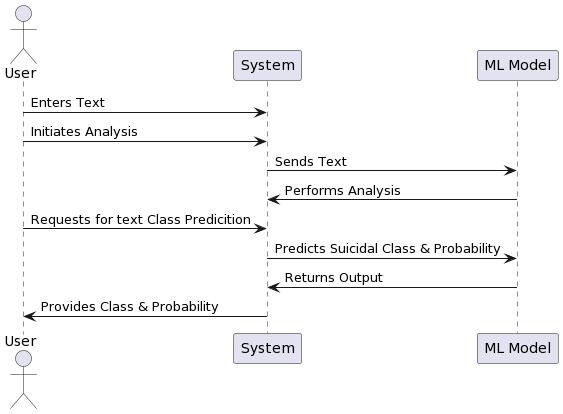


Fig 4.2.2 Sequence Diagram



DEPT OF CSE, GPREC, KNL **19** | P a g e

SUICIDAL THOUGHT DETECTION



**4.2.3 CLASS DIAGRAM**

The Class Diagram encapsulates the structural design of the suicide detection system, showcasing its essential components: User, System, and ML Model. It highlights their functionalities and interactions, illustrating the flow from text entry to predictive analysis and crucial output delivery. This diagram offers a comprehensive overview of the system's architecture and its crucial elements responsible for analyzing and predicting suicidal text.

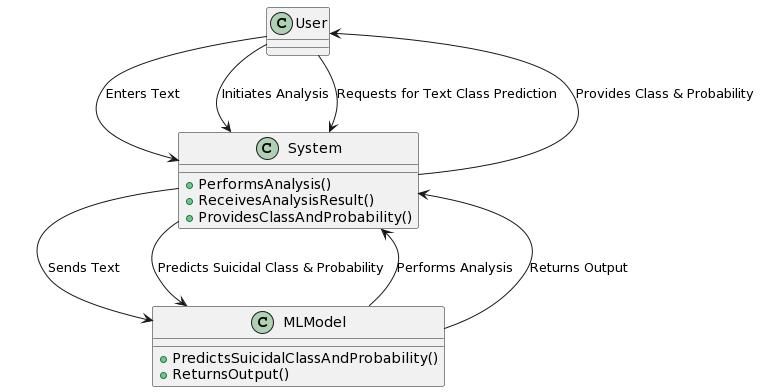


Fig 4.2.3 Class Diagram



DEPT OF CSE, GPREC, KNL **20** | P a g e

SUICIDAL THOUGHT DETECTION



**4.2.4 ACTIVITY DIAGRAM**

The Activity Diagram portrays the streamlined process within the suicide detection system, highlighting sequential steps from user input to the system's analysis through the ML model. It demonstrates text analysis for potential suicidal content, showing user interaction, system processing, and the crucial output of predicted class and probability, encapsulating the core workflow succinctly.

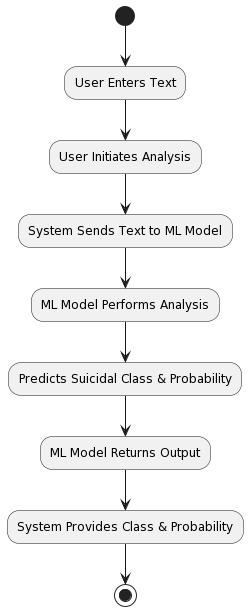


Fig 4.2.4 Activity Diagram



DEPT OF CSE, GPREC, KNL **21** | P a g e

SUICIDAL THOUGHT DETECTION



**4.3. IMPLEMENTATION**

In this section, we delve into the design and implementation details of the AI-Based Suicidal Thought Detection. The system is developed using the Flask web framework for the backend and integrates computer vision technologies. The implementation for the core functionalities is provided below.

**4.3.1 METHODS AND CATEGORIZATION**

Suicide detection has drawn the attention of many researchers due to an increasing suicide rate in recent years and has been studied extensively from many perspectives. The research techniques used to examine suicide also span many fields. The following figure (4.3.1) shows methods and categorization.

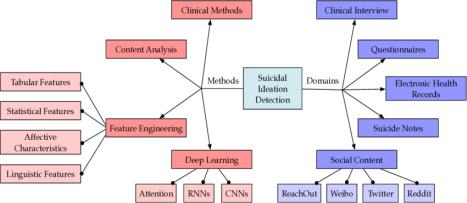


Fig. 4.3.1 Methods and Categorization

Some of the methods are:

**A. Content Analysis**

User's posts on social websites reveal rich information and their language preferences. Through exploratory data analysis on the user-generated content can have an insight into language usage and linguistic clues of suicide attempters.



DEPT OF CSE, GPREC, KNL **22** | P a g e

SUICIDAL THOUGHT DETECTION



**B. Feature Engineering**

The goal of text-based suicide classification is to determine whether candidates, through their posts, have suicidal ideations. Machine learning methods and NLP have also been applied in this field.

* 1. Tabular Features: Tabular data for suicidal ideation detection consist of questionnaire responses and structured statistical information extracted from websites. Such structured data can be directly used as features for classification or regression.
  2. General Text Features: Another direction of feature engineering is to extract features from unstructured text. The main features consist of N-gram features, knowledge-based features, syntactic features, context features, and class-specific features.
  3. Affective Characteristics: Affective characteristics are among the most distinct differences between those who attempt suicide and healthy individuals, which has drawn considerable attention from both computer scientists and mental health researchers.

1. **Deep Learning**

Deep learning has been a great success in many applications, including computer vision, NLP, and medical diagnosis. In the field of suicide research, it is also an important method for automatic suicidal ideation detection and suicide prevention. It can effectively learn text features automatically without sophisticated feature engineering techniques.

**4.3.2 PREREQUISITES**

Implementing a project for suicide ideation detection using machine learning models involves several prerequisites. Here's a brief overview:

**Understanding of Machine Learning**

Familiarity with machine learning concepts, including supervised learning, classification algorithms, and evaluation metrics.

**Programming Skills**

Proficiency in a programming language like Python, as many machine learning libraries (e.g., scikit-learn, TensorFlow, PyTorch) are commonly used in Python.

* **Data Collection and Preprocessing**
  + Knowledge of methods for collecting and preprocessing textual data, such as scraping social media posts, handling missing data, and text cleaning.



DEPT OF CSE, GPREC, KNL **23** | P a g e

SUICIDAL THOUGHT DETECTION



* **Text Representation Techniques**
  + Understanding of text representation techniques, including TF-IDF (Term Frequency-Inverse Document Frequency) and word embeddings, to convert text data into a format suitable for machine learning models.
* **Machine Learning Libraries**
  + Experience with popular machine learning libraries such as scikit-learn for implementing and training machine learning models.

**Model Selection and Evaluation**

Knowledge of different classification algorithms (e.g., Logistic Regression, Random Forest) and the ability to choose the most suitable model based on the problem requirements. Understanding the model evaluation metrics is essential.

**Ethical Considerations**

Awareness of ethical considerations related to mental health data, privacy, and responsible deployment of machine learning models.

**Interdisciplinary Collaboration**

Collaboration with mental health professionals, psychologists, or domain experts to ensure the ethical and responsible use of the model and to gain insights into the mental health context.

**Access to Data**

Access to a relevant dataset containing labeled examples of suicide ideation/non-suicide instances. Ensure compliance with data privacy regulations.

**Deployment Skills**

Basic knowledge of deploying machine learning models, whether it's on a web platform, in a mobile application, or as part of a larger system. Remember to approach the project with sensitivity and prioritize ethical considerations, given the nature of mental health data. Collaboration with experts in mental health is highly recommended to ensure a holistic and responsible approach to suicide ideation detection.



DEPT OF CSE, GPREC, KNL **24** | P a g e

SUICIDAL THOUGHT DETECTION



**4.3.3 PROJECT FILE STRUCTURE**

**Dataset Name:** Suicide Detection Dataset (SDD).

**Dataset Description:** The Suicide Detection Dataset (SDD) is a collection of textual and behavioral data aimed at training machine learning models to identify potential signs of suicidal ideation in individuals. This dataset is designed to aid in the development of algorithms that can automatically detect signs of distress, self-harm, or suicidal thoughts in online communication and behavioral patterns.

**Data Sources:** The various data sources for this project are

* Social Media Posts: A subset of publicly available social media posts (e.g., Twitter, Facebook, Reddit) where users discuss personal thoughts, emotions, and experiences related to mental health and well-being.
* Chat Logs: Textual data extracted from online chat platforms and forums where individuals communicate and seek support from peers or professionals.
* Behavioral Data: Information related to user behavior on online platforms, including posting frequency, time of activity, and engagement metrics.

**Data Categories:** The dataset is labelled to distinguish between suicidal and non-suicidal content.

Each data point is categorized as one of the following:

* Suicidal: Textual content or behavior that indicates thoughts of self-harm, suicide, or severe emotional distress.
* Non-suicidal: Textual content or behavior that does not indicate suicidal ideation.
* Textual Content: The text data containing user-generated content, which may include posts, comments, or chat messages.
* Behavioral Features: Features extracted from the user's online behavior, including posting frequency, time stamps, and engagement metrics.
* Label: A binary label indicating whether the content or behavior is suicidal or non-suicidal.
* Data Split: The dataset is divided into training, validation, and test sets to facilitate model development and evaluation.



DEPT OF CSE, GPREC, KNL **25** | P a g e

SUICIDAL THOUGHT DETECTION



**4.3.4 BUILDING THE PYTHON BASED PROJECT**

Build a Python-based suicide ideation detection project by defining project scope, collecting and preprocessing representative data, choosing and implementing classification models (e.g., Logistic Regression, Random Forest), and evaluating model performance. Prioritize ethical considerations, including user privacy and responsible data use. Optionally, create a user interface for interactive use, conduct thorough testing, deploy the model, and implement monitoring and updates. Collaborate with mental health experts throughout the project for a holistic understanding of the mental health context.

**4.3.5 PROJECT FLOW**

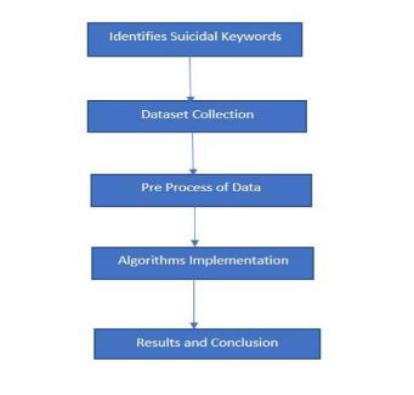


Figure 4.3.5 Project Flow Chart



DEPT OF CSE, GPREC, KNL **26** | P a g e

HEALTH DIAGNOSIS



**4.4 SOURCE CODE**

**4.4.1** **MACHINE LEARNING MODEL**

1. **Importing Libraries** *#Importing\_Libraries* import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.preprocessing import StandardScaler from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import classification\_report, confusion\_matrix, roc\_curve

from pickle import dump

%matplotlib inline

from google.colab import drive

drive.mount('/content/drive')

*#Reading dataset*

data = pd.read\_csv("/content/drive/MyDrive/kaggle/Suicide\_Detection.csv")

data.head()

import warnings

warnings.filterwarnings('ignore')

import numpy as np *# linear algebra*

import pandas as pd # data processing, CSV file I/O (e.g. pd.read\_csv)

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

from sklearn.feature\_extraction.text import TfidfVectorizer from sklearn.feature\_selection import SelectKBest,chi2,f\_classif from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import classification\_report , confusion\_matrix import pickle



DEPT OF CSE, GPREC, KNL **27** | P a g e

HEALTH DIAGNOSIS



import string

import nltk

from nltk.stem import PorterStemmer

import os

for dirname, \_, filenames in os.walk('/kaggle/input'):

for filename in filenames:

print(os.path.join(dirname, filename))

**b) Data Preprocessing**

Data pre-processing is a crucial step in preparing data for machine learning models. It

involves cleaning and transforming raw data into a format suitable for analysis. In the context of

suicidal thought detection or any natural language processing (NLP) task, data preprocessing

typically includes the following steps

* **Text Cleaning**

1. Remove irrelevant characters, symbols, and formatting issues. o Handle or remove special characters, numbers, and punctuation.
   1. Correct common typos and misspellings.

* **Tokenization**
  1. Break down text into smaller units, such as words or subwords (sub-tokenization). Tokenization is fundamental for understanding the structure of the text and extracting meaningful features.
* **Lowercasing**
  1. Convert all text to lowercase to ensure consistency and reduce the dimensionality of the data. This prevents the model from treating words with different cases as different features.
* **Stopword Removal**
  1. Eliminate common words (stopwords) that do not contribute much to the overall meaning of the text. This can include words like "and," "the," and "is."
* **Stemming and Lemmatization**
  1. Reduce words to their base or root form to consolidate similar words. Stemming involves removing suffixes, while lemmatization maps words to their base or dictionary form.



DEPT OF CSE, GPREC, KNL **28** | P a g e

HEALTH DIAGNOSIS



* **Handling Missing Data**
  1. Identify and handle missing or null values appropriately. Depending on the dataset, you may choose to impute missing values or remove instances with missing information.
* **Dealing with Imbalanced Data**
  1. If the dataset is imbalanced (i.e., one class is significantly underrepresented), consider techniques such as oversampling the minority class or under sampling the majority class to balance the distribution.
* **Encoding Categorical Variables**
  1. Convert categorical variables into numerical representations using techniques like one-hot encoding or label encoding.
* **Feature Engineering**
  1. Extract relevant features from the text data, such as sentiment scores, word embeddings, or other linguistic features that capture the context and emotion of the text.
* **Data Splitting**
  1. Split the dataset into training, validation, and testing sets to assess the model's performance on unseen data.
* **Handling Biases**
  1. Be aware of biases in the data that may affect model predictions. Take steps to address biases ethically, ensuring fair and unbiased model outcomes.

*#DataPreprocessing*

data.shape

df = data.sample(n=10000, random\_state=42)

df.info()

df['Unnamed: 0'].is\_unique

df.drop(columns = 'Unnamed: 0',inplace=True)

df.isnull().sum()

df.duplicated().sum()

*#Data Visualization*

classCnt = df['class'].value\_counts()

print(classCnt)

plt.figure(figsize = ((20,5)))



DEPT OF CSE, GPREC, KNL **29** | P a g e

HEALTH DIAGNOSIS



plt.subplot(1,2,1)

sns.countplot(df,x='class')

plt.subplot(1,2,2)

plt.pie(classCnt,labels = classCnt.index,autopct='%.0f%%')

plt.show()

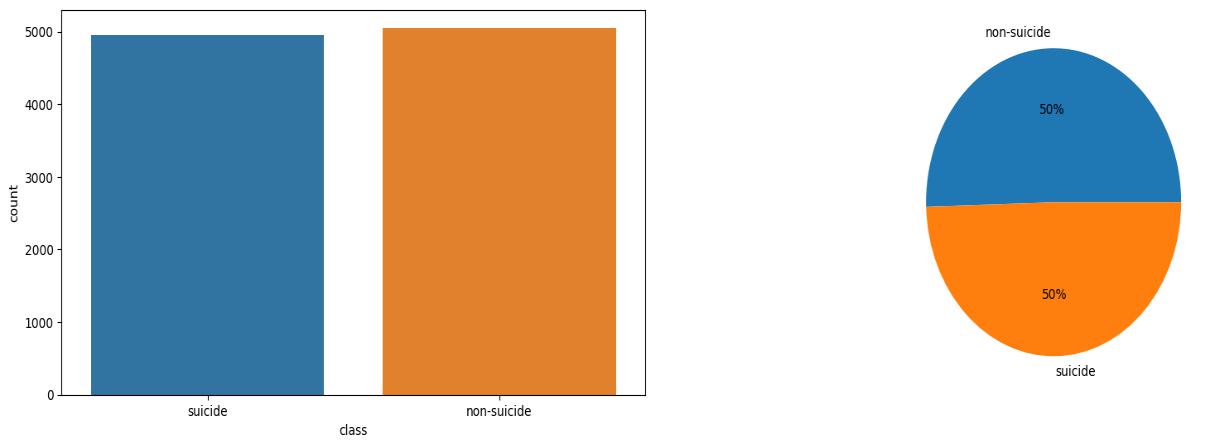


Fig. 4.4.1 Data Visualization

*#Lowering The Text*

df['text'] = df['text'].str.lower()

*#Remove Punctuation*

df['text'] = df['text'].str.replace(r'[^\w\s]+', '',regex = True)

import nltk

nltk.download('stopwords')

*#Stop word removal*

from nltk.corpus import stopwords

stop\_words = stopwords.words('english')

df['text'] = df['text'].apply(lambda x: ' '.join([word for word in x.split() if word not in

(stop\_words)]))

#import nltk

nltk.download('punkt')

*#Tokenization*



DEPT OF CSE, GPREC, KNL **30** | P a g e

HEALTH DIAGNOSIS



df['text'] = df['text'].apply(lambda x:nltk.word\_tokenize(x))

*#Stemming*

ps = PorterStemmer()

df['text'] = df['text'].apply(lambda x : [ps.stem(i) for i in x])

df['text']=df['text'].apply(lambda x : ' '.join(x))

df.head()

*#Saved the cleaned dataset*

df.to\_csv('file1.csv')

df2 = pd.read\_csv('file1.csv')

df2.head()

df2.info()

df.iloc[ind]

df2.dropna(inplace=True)

* x,y = df2['text'],df2['class'] x=df2['text']

y = df2['class'].map({'suicide': 1, 'non-suicide': 0})

**c) Feature Extraction**

Feature extraction is a critical step in preparing data for machine learning models, particularly in the context of suicidal thought detection. In natural language processing (NLP), features are representations of the input text that the model uses for learning and making predictions. Here are some common techniques for feature extraction in suicidal thought detection.

* **TF-IDF (Term Frequency-Inverse Document Frequency)**

Similar to BoW, TF-IDF assigns weights to words based on their frequency in a document

relative to their frequency in the entire corpus. It helps prioritize words that are more informative for a specific document.

* **Sentiment Analysis**

Extract sentiment scores or features that capture the emotional tone of the text. Sentiment

analysis tools can assign positive, negative, or neutral labels to sentences or documents.



DEPT OF CSE, GPREC, KNL **31** | P a g e

HEALTH DIAGNOSIS



* **Length and Structure**

Include features related to the length of the text, such as the number of words or sentences.

Additionally, consider features related to the structural elements of the text, such as the use of paragraphs or line breaks.

It's important to experiment with different feature extraction techniques and evaluate their impact on the model's performance. Additionally, combining multiple types of features can often lead to more robust representations. Regularly revisiting and updating feature extraction methods based on the characteristics of the data and the task can contribute to improved model accuracy and sensitivity to suicidal thought detection.

#TF-IDF vectorizer(Term Frequency Inverse Document frequency)

# min\_df=50

vectorizer = TfidfVectorizer(max\_features=5000)

x = vectorizer.fit\_transform(x)

*# Save the model*

#with open('tfidf.pkl', 'wb') as f:

# pickle.dump(vectorizer, f)

X\_train,X\_test,y\_train,y\_test = train\_test\_split(x,y,test\_size=0.30,random\_state=3) X\_train.shape,X\_test.shape

1. **About The Algorithms used**
   * **Logistic Regression**

Logistic Regression for suicidal thought detection is a binary classification algorithm that models the probability of suicidal thoughts based on input features. It estimates the likelihood of an instance belonging to the positive class, making it suitable for analysing textual data and predicting the presence of suicidal thoughts.

*#Logistic Regression*

from sklearn.linear\_model import LogisticRegression from sklearn.metrics import confusion\_matrix



DEPT OF CSE, GPREC, KNL **32** | P a g e

HEALTH DIAGNOSIS



from sklearn.metrics import accuracy\_score ,precision\_score,recall\_score,f1\_score

log = LogisticRegression()

log.fit(X\_train,y\_train)

print('Training score:',log.score(X\_train, y\_train)\*100)

print('Testing score:',log.score(X\_test,y\_test)\*100)

user\_input = input()

user\_input\_vector = vectorizer.transform([user\_input])

prediction = log.predict(user\_input\_vector)

print(prediction)

if prediction[0] == 1:

print("The input belongs to the 'suicide' class.")

else:

print("The input belongs to the 'non-suicide' class.")

#Probability

probabilities = log.predict\_proba(user\_input\_vector) probability\_non\_suicide = probabilities[0][0] probability\_suicide = probabilities[0][1]

print('Probability of non-suicide:', probability\_non\_suicide \* 100, '%')

print('Probability of suicide:', probability\_suicide \* 100, '%')

if probability\_suicide > probability\_non\_suicide:

print("The input is more likely to belong to the 'suicide' class.")

else:

print("The input is more likely to belong to the 'non-suicide' class.")

**Output**

Training score: 93.5419345620803

Testing score: 91.3

I can't take it anymore. Life is a lie

[1]

The input belongs to the 'suicide' class.



DEPT OF CSE, GPREC, KNL **33** | P a g e

HEALTH DIAGNOSIS



Probability of non-suicide: 14.707589540386723 %

Probability of suicide: 85.29241045961328 %

The input is more likely to belong to the 'suicide' class.

* **Naive Bayes**

Naive Bayes for suicidal thought detection is a probabilistic classification algorithm that assumes feature independence given the class. It's well-suited for analyzing text data by leveraging word occurrences and frequencies, making it efficient and effective in identifying potential indicators of suicidal thoughts in written content**.**

*#Naive Bayes*

from sklearn.naive\_bayes import MultinomialNB

* Replace Logistic Regression with Naive Bayes naive\_bayes = MultinomialNB() naive\_bayes.fit(X\_train, y\_train)

print('Training score:', naive\_bayes.score(X\_train, y\_train) \* 100) print('Testing score:', naive\_bayes.score(X\_test, y\_test) \* 100)

**Output**

Training score: 90.98442634662095

Testing score: 88.03333333333333

* **Random Forest**

Random Forest algorithm for suicidal thought detection leverages an ensemble of decision trees, each trained on a random subset of features and data. By combining predictions through voting, it enhances robustness and accuracy, making it effective in analysing linguistic patterns and emotional content to identify potential indicators of suicidal thoughts in text data.

*#Random Forest*

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score, classification\_report

# Initialize the Random Forest classifier

random\_forest = RandomForestClassifier(n\_estimators=100, random\_state=42)

# Train the Random Forest model



DEPT OF CSE, GPREC, KNL **34** | P a g e

HEALTH DIAGNOSIS



random\_forest.fit(X\_train, y\_train)

# Make predictions on the training set

train\_predictions = random\_forest.predict(X\_train)

# Make predictions on the testing set

test\_predictions = random\_forest.predict(X\_test)

# Evaluate the Random Forest model

train\_accuracy = accuracy\_score(y\_train, train\_predictions)

test\_accuracy = accuracy\_score(y\_test, test\_predictions)

print('Random Forest Training Accuracy:', train\_accuracy \* 100)

print('Random Forest Testing Accuracy:', test\_accuracy \* 100)

* Classification report for detailed evaluation print('\nRandom Forest Classification Report:') print(classification\_report(y\_test, test\_predictions))

**Output**

Random Forest Training Accuracy: 99.95713673381911

Random Forest Testing Accuracy: 88.26666666666667

Random Forest Classification Report

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | precision | recall |  | f1-score | support |
| 0 | 0.88 | 0.89 |  | 0.89 | 1538 |
| 1 | 0.88 | 0.88 |  | 0.88 | 1462 |
|  | accuracy |  |  | 0.88 | 3000 |
|  | macro avg | 0.88 | 0.88 | 0.88 | 3000 |
| weighted avg | | 0.88 | 0.88 | 0.88 | 3000 |

* **Decision Tree Classifier**

Decision Tree Classifier for suicidal thought detection use a hierarchical structure to make decisions based on input features, recursively partitioning data. They analyse textual content, capturing linguistic patterns and emotional cues, making them effective in identifying potential signs of suicidal thoughts through interpretable decision paths.

*#Decision Tree Classifier*



DEPT OF CSE, GPREC, KNL **35** | P a g e

HEALTH DIAGNOSIS



from sklearn.tree import DecisionTreeClassifier

* Replace Logistic Regression with Decision Tree decision\_tree = DecisionTreeClassifier() decision\_tree.fit(X\_train, y\_train)

print('Training score:', decision\_tree.score(X\_train, y\_train) \* 100) print('Testing score:', decision\_tree.score(X\_test, y\_test) \* 100)

**Output**

Training score: 99.95713673381911

Testing score: 83.39999999999999

* **Support Vector Machine**

Support Vector Machines (SVM) for suicidal thought detection is a machine learning algorithm that aims to find an optimal hyperplane to separate data into classes. SVM analyses text features, effectively capturing complex patterns, and is suitable for identifying nuanced linguistic cues indicative of potential suicidal thoughts in textual content.

*#Support Vector Machine*

from sklearn.svm import SVC

* Replace Logistic Regression with Support Vector Machine svm\_classifier = SVC(probability=True) svm\_classifier.fit(X\_train, y\_train)

print('Training score:', svm\_classifier.score(X\_train, y\_train) \* 100) print('Testing score:', svm\_classifier.score(X\_test, y\_test) \* 100)

* User input

user\_input = input()

user\_input\_vector = vectorizer.transform([user\_input])

prediction = svm\_classifier.predict(user\_input\_vector)

print(prediction)

if prediction[0] == 1:

print("The input belongs to the 'suicide' class.")

else:



DEPT OF CSE, GPREC, KNL **36** | P a g e

HEALTH DIAGNOSIS



print("The input belongs to the 'non-suicide' class.")

# Probability

probabilities = svm\_classifier.predict\_proba(user\_input\_vector) probability\_non\_suicide = probabilities[0][0] probability\_suicide = probabilities[0][1]

print('Probability of non-suicide:', probability\_non\_suicide \* 100, '%')

print('Probability of suicide:', probability\_suicide \* 100, '%')

if probability\_suicide > probability\_non\_suicide:

print("The input is more likely to belong to the 'suicide' class.")

else:

print("The input is more likely to belong to the 'non-suicide' class.")

**Output**

Training score: 98.52836119445635

Testing score: 91.60000000000001

Feeling blessed today.

[0]

The input belongs to the 'non-suicide' class.

Probability of non-suicide: 88.68077279410409 %

Probability of suicide: 11.319227205895919 %

The input is more likely to belong to the 'non-suicide' class.

**4.4.2 WEB INTERFACE**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>Is Your Text Suicidal?</title>**

**<!-- Include Chart.js library -->**

**<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>**

**<link** **href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"**

**rel="stylesheet"** **integrity="sha384-**



DEPT OF CSE, GPREC, KNL **37** | P a g e

HEALTH DIAGNOSIS



**T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN"**

**crossorigin="anonymous">**

**</head>**

**<style>**

**body {**

**background-repeat: no-repeat;**

**background-size: cover;**

**background-image: url("{{ url\_for('static', filename='images/BGIMG.jpg') }}");**

**margin-left: 800px;**

**}**

**</style>**

**<body>**

**<br><br>**

**<h1>Is Your Text Suicidal?</h1>**

**<form method="POST" action="#">**

**<label for="text"><h2>Enter your text:</h2></label> <br>**

**<textarea id="text" name="text" rows="4" cols="50"></textarea> <br>**

**<input type="submit" value="Predict" class="btn btn-secondary small"> </form>**

**<!-- {{data}} -->**

**<div style="display: flex;">**

**<div style="width: 400px; margin: 0; margin-top: 20px;"> <canvas id="myPieChart"></canvas>**

**</div>**

**<!-- {{data}} -->**

**<div >**

**{% if data is defined %}**

**{% if data[0]==1 is defined %}**



DEPT OF CSE, GPREC, KNL **38** | P a g e

HEALTH DIAGNOSIS



**<img src="{{url\_for('static',filename='images/SC1.png')}}" alt="" width="600px"> {% else %}**

**<img src="{{url\_for('static',filename='images/NSC1.png')}}" alt="">**

**{%endif%}**

**{% endif %}**

**</div>**

**</div>**

**<script>**

* **Data for the pie chart const data = {**

**{% if data is defined %}**

**labels: [**

**'NON-SUICIDE CLASS',**

**'SUICIDE CLASS'**

**],**

**{%endif%}**

**datasets: [{**

**data: [**

**{% if data is defined %}**

**{{ data[1] }} , {{ data[2] }} {% else %}**

**0, 0**

**{% endif %}**

**],**

**backgroundColor: [**

**'rgb(255, 99, 132)',**

**'rgb(54, 162, 235)'**

**],**



DEPT OF CSE, GPREC, KNL **39** | P a g e

HEALTH DIAGNOSIS



**hoverOffset: 4**

**}]**

**};**

**// Get the canvas element**

**var ctx = document.getElementById('myPieChart').getContext('2d'); var myPieChart = new Chart(ctx, {**

**type: 'pie',**

**data: data,**

**});**

* **Create the pie chart </script>**

**<script**

**src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"**

**integrity="sha384-**

**C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"**

**crossorigin="anonymous"></script>**

**</body>**

**</html>**

**4.4.2 FLASK INTEGRATION**

**a)** **Flask Setup**

 The Flask web framework is imported, and an instance of the Flask app is created with

app = Flask(\_\_name\_\_).

 A route is defined using @app.route('/'), specifying that both GET and POST requests to

the root URL ('/') should be handled by the following function.

**b) User Input Handling**

* The home() function is defined to handle requests to the root URL.
* Within this function, the request object is used to check if the incoming request is a POST request (indicating that the form has been submitted).



DEPT OF CSE, GPREC, KNL **40** | P a g e

HEALTH DIAGNOSIS



**c)** **HTML Template Rendering**

* The render\_template function is used to render the 'index.html' template.
* The prediction results are passed to the template as the data variable, which can be accessed in the HTML file for display.

**d) Run the Flask Application**

* The script checks if it's the main module (\_\_name\_\_=="\_\_main\_\_") and runs the Flask app in debug mode using app.run(debug=True).

import pandas as pd

from flask import Flask,render\_template,request from sklearn.model\_selection import train\_test\_split

from sklearn.feature\_extraction.text import TfidfVectorizer

df2=pd.read\_csv("C:\\Users\\home\\Desktop\\STD\\file1.csv")

df2.dropna(inplace=True)

x=df2['text']

1. = df2['class'].map({'suicide': 1, 'non-suicide': 0}) vectorizer = TfidfVectorizer(max\_features=5000) x = vectorizer.fit\_transform(x)

X\_train,X\_test,y\_train,y\_test = train\_test\_split(x,y,test\_size=0.30,random\_state=3)

from sklearn.linear\_model import LogisticRegression from sklearn.metrics import confusion\_matrix

from sklearn.metrics import accuracy\_score ,precision\_score,recall\_score,f1\_score

log = LogisticRegression()

log.fit(X\_train,y\_train)

app=Flask(\_\_name\_\_)

@app.route('/',methods=['POST','GET'])



DEPT OF CSE, GPREC, KNL **41** | P a g e

HEALTH DIAGNOSIS



def home():

if request.method=="POST":

# text=request.form['text']

user\_input = request.form['text']

if(len(user\_input)!=0):

user\_input\_vector = vectorizer.transform([user\_input])

prediction = log.predict(user\_input\_vector)

if prediction[0]==1:

flag=1

else:

flag=0

probabilities = log.predict\_proba(user\_input\_vector) probability\_non\_suicide = probabilities[0][0] probability\_suicide = probabilities[0][1] PN=round(probability\_non\_suicide \* 100) PS=round(probability\_suicide \* 100)

return render\_template("index.html",data=[flag,PN,PS])

else:

return render\_template("index.html")

return render\_template("index.html")

if \_\_name\_\_=="\_\_main\_\_":

app.run(debug=True)



DEPT OF CSE, GPREC, KNL **42** | P a g e

HEALTH DIAGNOSIS



**4.5 OUTPUTS**

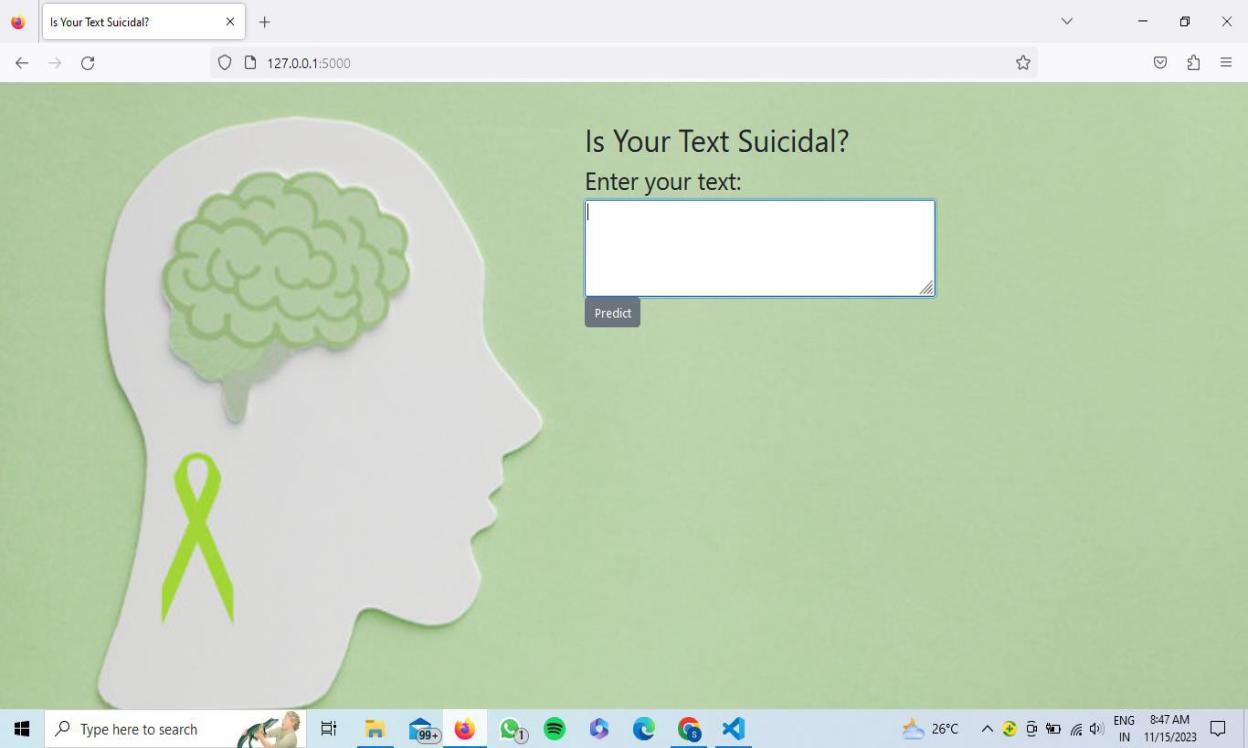


Fig 4.5.1 Web Interface



DEPT OF CSE, GPREC, KNL **43** | P a g e

HEALTH DIAGNOSIS

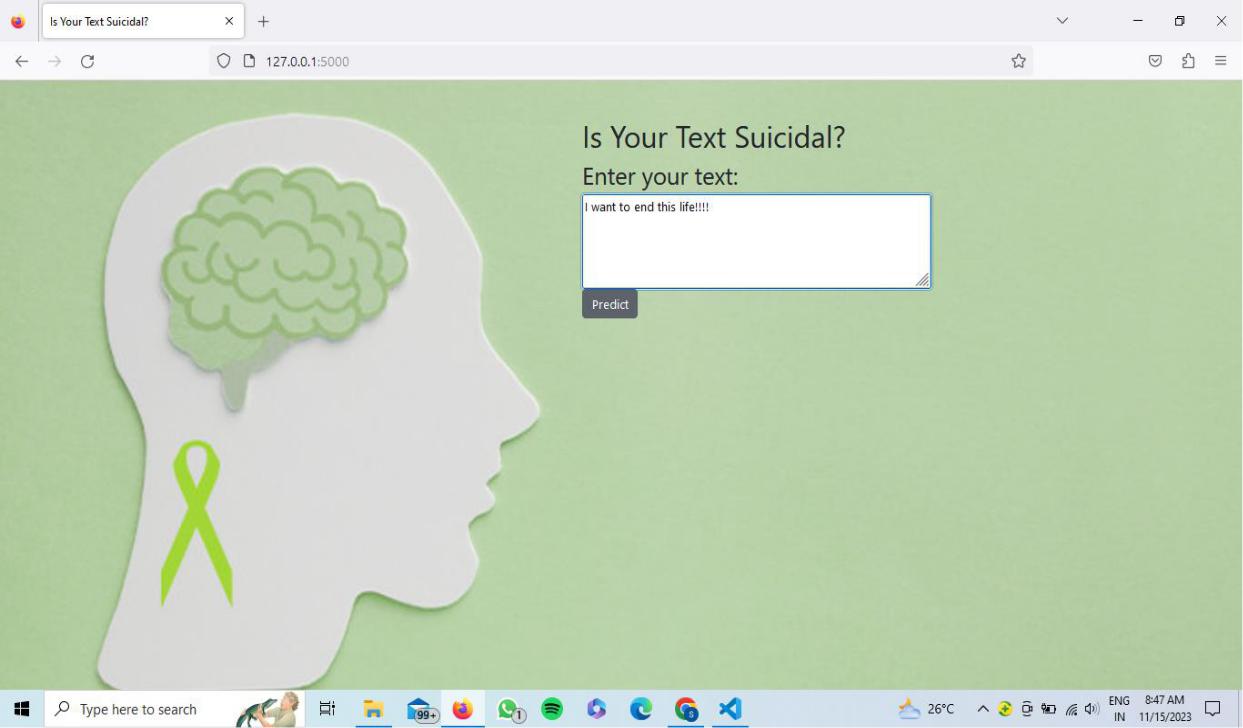


Fig 4.5.2 Input - Suicidal Text

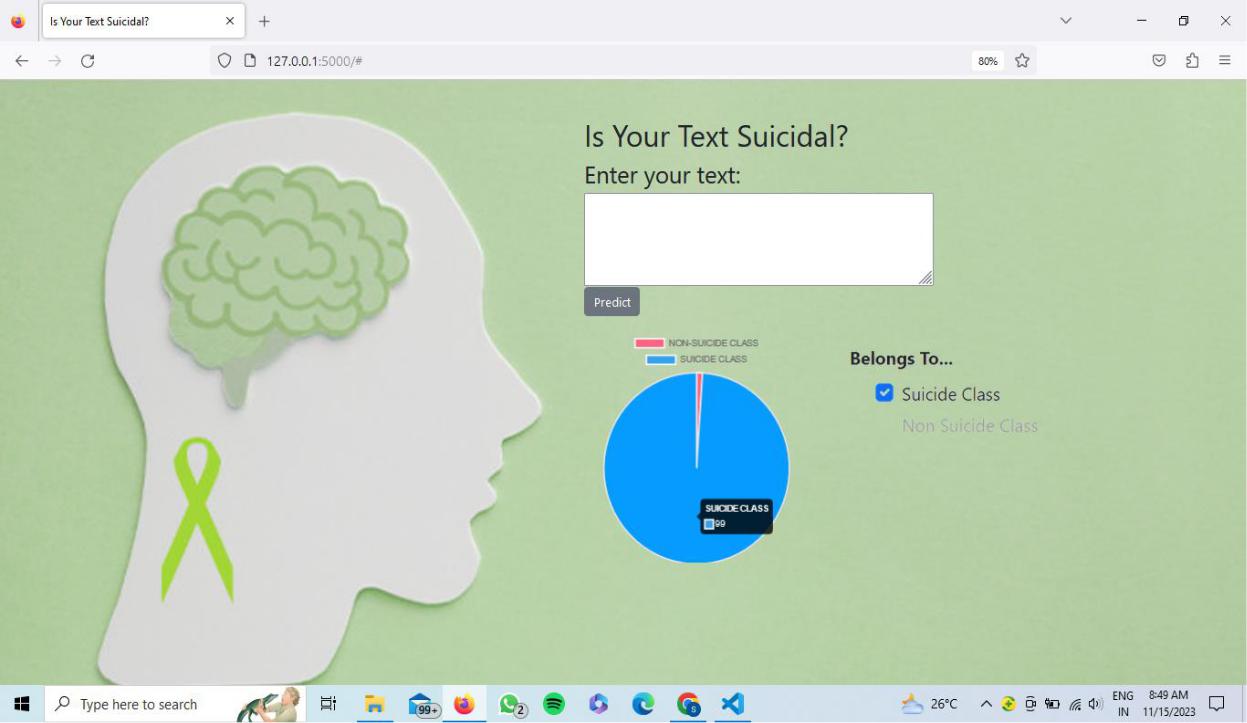


Fig 4.5.3 Suicidal Class Prediction



DEPT OF CSE, GPREC, KNL **44** | P a g e

HEALTH DIAGNOSIS

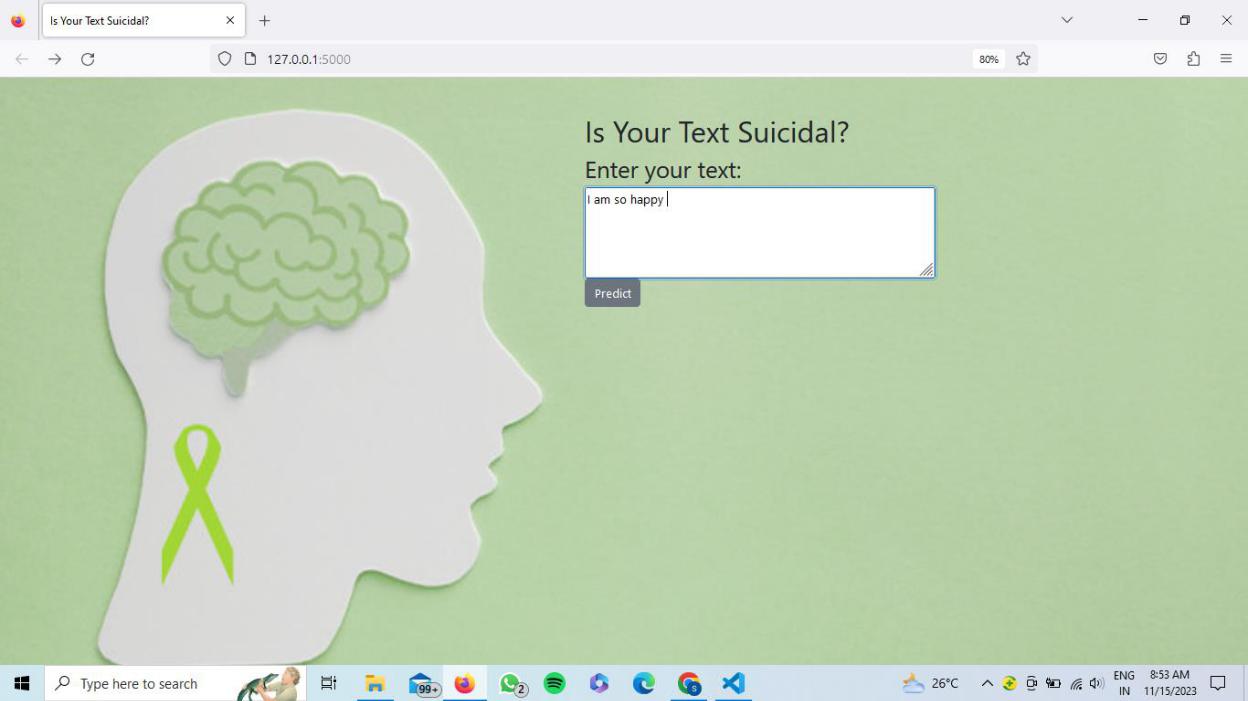


Fig 4.5.4 Input – Non-Suicidal Text

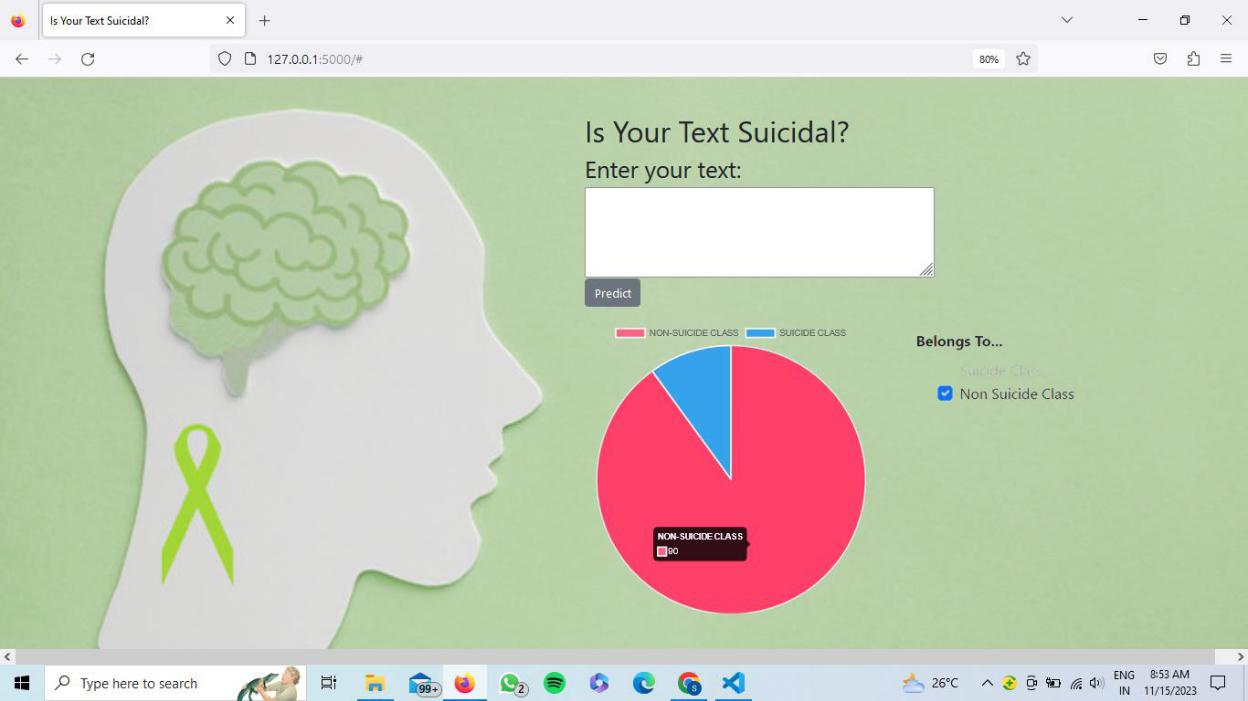


Fig 4.5.5 Non-Suicidal Class Prediction



DEPT OF CSE, GPREC, KNL **45** | P a g e

HEALTH DIAGNOSIS



**CONCLUSION**



DEPT OF CSE, GPREC, KNL

HEALTH DIAGNOSIS



**5. CONCLUSION**

**5.1 CONCLUSION**

In conclusion, this project successfully demonstrates the application of machine learning techniques in the early detection and prediction of **Diabetes**, **Chronic Kidney Disease (CKD)**, and **Heart Disease**. Using algorithms like Logistic Regression, Random Forest, and K-Nearest Neighbors (KNN), we were able to analyze medical data from CSV datasets to build accurate and efficient models. Python, along with Flask, was used for backend development, while HTML, CSS, and Bootstrap provided a user-friendly interface for web deployment. The models achieved encouraging accuracy scores, showing the potential of technology to support healthcare professionals in early diagnosis and treatment planning. Overall, this project contributes to raising awareness about the role of data science in preventive healthcare.

**5.2 FUTURE ENHANCEMENT**

The future enhancement of the Suicidal Thought Detection project involves incorporating real-time data from various social media platforms. This expansion aims to detect user texts that may trigger suicidal thoughts, providing timely intervention and support. The proposed steps for this enhancement include:

1. **Real-Time Data Integration**
   * Implement mechanisms to continuously gather and analyze real-time data from popular social media platforms such as Kaggle.

**2.Multidisease Prediction System**

* Extend the system to support detection of more diseases in a single unified platform using a multi-label classification approach.

**3.Advanced Model Optimization**

* Implement deep learning models or ensemble techniques for better accuracy and generalization across larger datasets.



DEPT OF CSE, GPREC, KNL **46** | P a g e



**4.Educational Resources**

* + Integrate educational resources within the project to provide users with information on mental health support and resources, fostering a holistic approach to well-being



DEPT OF CSE, GPREC, KNL **47** | P a g e

HEALTH DIAGNOSIS



**REFERENCES**

1.Diabetes Prediction using Machine Learning Algorithms

Available at [DIABETES PREDICTION USING ML ALGORITHMS](https://www.sciencedirect.com/science/article/pii/S1877050920300557)

2 Diabetes prediction using machine learning and explainable AI techniques

Available at: [Diabetes prediction using machine learning and explainable AI techniques](https://pmc.ncbi.nlm.nih.gov/articles/PMC10107388/)

3. Chronic kidney disease prediction based on machine learning algorithms

Available at: [Chronic kidney disease prediction based on machine learning algorithms](https://pmc.ncbi.nlm.nih.gov/articles/PMC9874070/)

4. Chronic Kidney Disease Prediction Using Machine Learning Methods

Available at: <https://ieeexplore.ieee.org/document/9185249>

5. Prediction of Heart Disease Based on Machine Learning

Available at: <https://pmc.ncbi.nlm.nih.gov/articles/PMC10378171/>

6.Heart Disease Prediction Using Machine Learning

Available at: <https://ieeexplore.ieee.org/document/9734880>





DEPT OF CSE, GPREC, KNL **48** | P a g e

HEALTH DIAGNOSIS



DEPT OF CSE, GPREC, KNL **49** | P a g e