



# **CLASSIFYING DEPRESSION USING MACHINE LEARNING ON SOCIAL MEDIA TWEETS**



**A PROJECT REPORT**

*Submitted by*

**BIPIN V K (710719205010)**

**CRIS CUMINS P (710719205012)**

**WILSON A (710719205058)**

*in partial fulfilment for the award of the degree*

*of*

**BACHELOR OF TECHNOLOGY**

**IN**

**INFORMATION TECHNOLOGY**

**Dr. N.G.P. INSTITUTE OF TECHNOLOGY, COIMBATORE – 641048**

**(AN AUTONOMOUS INSTITUTION)**

**ANNA UNIVERSITY: CHENNAI 600 025**

**APRIL 2023**



**CLASSIFYING DEPRESSION USING  
MACHINE LEARNING  
ON SOCIAL MEDIA TWEETS**



**A PROJECT REPORT**

*Submitted by*

**BIPIN V K (710719205010)**

**CRIS CUMINS P (710719205012)**

**WILSON A (710719205058)**

*in partial fulfilment for the award of the degree*

*of*

**BACHELOR OF TECHNOLOGY**

**IN**

**INFORMATION TECHNOLOGY**

**Dr. N.G.P. INSTITUTE OF TECHNOLOGY, COIMBATORE – 641048**

**(AN AUTONOMOUS INSTITUTION)**

**ANNA UNIVERSITY :: CHENNAI 600 025**

**APRIL 2023**

## **BONAFIDE CERTIFICATE**

Certified that this project report “**CLASSIFYING DEPRESSION USING MACHINE LEARNING ON SOCIAL MEDIA TWEETS**” is the Bonafide work of “**BIPIN V K (710719205010), CRIS CUMINS P (710719205012), WILSON A (710719205058)**” who carried out the project work under my supervision.

### **SIGNATURE**

**Dr. M. KRISHNAMOORTHY M.E., Ph.D.,**

**HEAD OF THE DEPARTMENT**

Department of Information Technology,  
Dr. N. G. P Institute of Technology,  
Kalapatti Road,  
Coimbatore-641 048.

### **SIGNATURE**

**Ms. BIJI ROSE M.E.,**

**SUPERVISOR**

Assistant Professor (SG),  
Department of Information Technology,  
Dr. N.G.P Institute of Technology,  
Kalapatti Road,  
Coimbatore-641 048.

Submitted for the End Semester Project Viva-Voce held on \_\_\_\_\_

-----

**INTERNAL EXAMINER**

-----

**EXTERNAL EXAMINER**

## ACKNOWLEDGEMENT

We extend our heartiest thanks to **Dr. NALLA G. PALANISWAMI**, Chairman, KMCH & Dr. N.G.P. Educational Institutions for providing us the necessary infrastructure to do our project work.

We express our gratitude to **Dr. THAVAMANI D. PALANISWAMI**, Secretary, Dr. N.G.P. Institute of Technology, for providing us the facilities to do our project work.

We would like to express our hearty thanks and gratitude to **Dr. S.U. PRABHA, M.E., Ph.D.**, Principal, Dr. N.G.P. Institute of Technology, for her earnest encouragement.

We extend our deep sense of gratitude to **Dr. M. KRISHNAMOORTHY, M.E., Ph.D.**, Head of the Department, Department of Information Technology, for his valuable guidance and constructive suggestion at all stages of our project from inception to completion.

We express our hearty thanks to our project guide **Ms. BIJI ROSE, M.E.**, Assistant Professor (SG), Department of Information Technology, for her valuable guidance and timely help for completing our project.

We express our sincere thanks to our project coordinator **Dr. R.P. NARMADHA, M.E., Ph.D.**, Assistant Professor (SG), Department of Information Technology, for her support in developing our project.

We would also like to express our gratitude to the faculty members of Department Information Technology and also to our family for their kind patronage.

## **ABSTRACT**

Human psychological depression and human emotion are very much interconnected. In computational psychology study, the relationship between depression and emotions is the key to understanding the human behavior. Many research has been done for detecting Depression from image, sound, health etc. Using many algorithms but has not been explicitly taken up yet to find psychological depression. In this project, a Machine learning data classification system is presented in which Support Vector Machine (SVM) and K-Nearest Neighbor (KNN) is used for data classification. By employing the dataset, this project examines and identifies people's levels of psychological depression from social media tweets. In order to obtain the psychological depression state of unknown individuals, various text that are associated with psychological depression have also been found.

**KEYWORDS:** Depression, social media, Machine learning, KNN, SVM.

## **TABLE OF CONTENTS**

<b>CHAPTER NO.</b>	<b>TITLE</b>	<b>PAGE NO</b>
	<b>ABSTRACT</b>	
	<b>LIST OF FIGURES</b>	
	<b>LIST OF ABBREVIATION</b>	
<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
	1.1 OBJECTIVE	3
<b>2</b>	<b>LITERATURE SURVEY</b>	<b>5</b>
	2.1 INTRODUCTION	5
	2.2 LITERATURE REVIEW	5
<b>3</b>	<b>SYSTEM ANALYSIS</b>	<b>10</b>
	3.1 EXISTING SYSTEM	10
	3.1.1 Drawbacks of Existing System	11
	3.2 PROPOSED SYSTEM	12
<b>4</b>	<b>SYSTEM SPECIFICATION</b>	<b>14</b>
	4.1 INTRODUCTION	14
	4.2 HARDWARE SPECIFICATION	14
	4.3 SOFTWARE SPECIFICATION	14
	4.4 SOFTWARE DESCRIPTION	14
	4.4.1 Python Programming Language	14

	4.4.2 Python 3.7	15
	4.4.3 Google Colab	17
	4.4.4 Modules and Packages	17
<b>5</b>	<b>SYSTEM OVERVIEW</b>	<b>21</b>
	5.1 MODULE DESCRIPTION	21
	5.1.1 Dataset	21
	5.1.2 SVM Classifier	21
	5.1.3 KNN Classifier	24
	5.1.4 Preprocessing	27
	5.1.5 Model Selection	27
	5.1.6 Performance Evaluation Matrix	27
	5.1.7 Generalization	28
	5.1.8 Confusion Matrix	29
	5.1.9 Classified Result	30
<b>6</b>	<b>SYSTEM IMPLEMENTATION</b>	<b>33</b>
	6.1 SYSTEM IMPLEMENTATION	33
	6.1.1 Data Preparation	34
	6.1.2 Model Comparison	34
	6.1.3 Differentiation	34
<b>7</b>	<b>RESULT AND OUTPUT</b>	<b>35</b>
	7.1 EXPERIMENTAL RESULT	35

	7.1.1 Accuracy Rate Of SVM	35
	7.1.2. Accuracy Rate Of KNN	36
	7.2 OUTPUT	36
<b>8</b>	<b>CONCLUSION AND FUTURE WORK</b>	<b>38</b>
	8.1 CONCLUSION	38
	8.2 FUTURE WORK	38
	<b>REFERENCES</b>	



## LIST OF TABLES

TABLE NO	TABLE NAME	PAGE NO
1	Support Vector Machine Accuracy	30
2	K-nearest Neighbors Accuracy	31

## **LIST OF FIGURES**

<b>FIGURE NO.</b>	<b>FIGURE NAME</b>	<b>PAGE NO.</b>
<b>1</b>	Block Diagram	12
<b>2</b>	Flow Diagram of SVM	23
<b>3</b>	Flow Diagram of KNN	24
<b>4</b>	KNN Dataset Plotted	26
<b>5</b>	Classify new data point	26
<b>6</b>	Confusion Matrix of SVM	29
<b>7</b>	Confusion Matrix of KNN	30
<b>8</b>	Accuracy Rate of SVM	35
<b>9</b>	Accuracy Rate of KNN	36
<b>10</b>	Output of NORMAL Prediction	32
<b>11</b>	Output of MILD DEPRESSED Prediction	33
<b>12</b>	Output of DEPRESSION Prediction	33

## **LIST OF ABBREVIATION**

BRD	Bovine Respiratory Disease
ECG	Electrocardiogram
GSR	Galvanic Skin Response
HMM	Hidden Markov Models
KNN	K-Nearest Neighbor
MPM	Maximum Posterior Marginal
SVM	Support Vector Machine
PCP	Principal Component Process
PS	Psychological Depression
PCA	Principal Component Analysis

# **CHAPTER – 1**

## **INTRODUCTION**

Modern life involves more labor than leisure. The health consequences of such a circumstance could be dire. According to a study by the Indian Psychological Association, 48% of Indians believe that their depression has gotten worse over the previous five years, and 33% of Indians report having extreme depression. For these findings to be established universally, more research is needed. In general, we believe that people are becoming more vulnerable to illness because they are experiencing more depression in their daily lives than ever before. Depression has both physical and social consequences. Long-held beliefs hold that the impacts of depressive disorders can result in poor health, strained interpersonal relationships, and decreased productivity at work. It was typically linked to significant health-related issues in 2009. Recent research in cattle models has established that psychological depression has a substantial impact on the cause and prognosis of bovine respiratory disease and that PS-related biomarkers can be used to predict the diagnosis of BRD. People affected by major depressive Disorder indicates that low-income countries have a higher percentage of depressed individuals, which has an impact on the likelihood that these individuals will receive professional therapy. It is obvious that there is a pressing need to fight depression by quickly recognizing hazards through the usage of social media platforms. Finding and helping people who are sad can be made possible by identifying them online who lack access to professional treatment because of the hurdles.

Psychological analysis is a process in which psychological data are extracted from text-based data. To eliminate emotions, opinions, and judgement-forming, text data are used. Having opinions or views toward some products or any topic is human psychology, which defines what one thinks about the products or topic. Nowadays, the way of expressing various emotions and giving opinions has changed drastically

with the advancement of social media and Internet technology. People use blogs, product recommendation and review websites, and other social media to give opinions about products, movies, and political parties and on current important topics. Famous social media platforms such as Facebook, Twitter, and Reddit have become the most reliable platforms for sharing opinions and reviews among a new generation of Internet users. Business firms and organizations use people-oriented psychological feedback to increase their products' value and quality.

Depression is a significant and growing public health concern, affecting millions of people worldwide. It is a complex and multifaceted disorder that can be difficult to diagnose and treat. In recent years, there has been growing interest in using machine learning and natural language processing techniques to identify and classify depression from social media data, such as tweets. Social media platforms like Twitter provide a unique opportunity to collect large amounts of data that can be used to better understand depression and its symptoms, as well as to develop more effective treatments.

The aim is to classify depression using machine learning algorithms applied to social media tweets. We will use a dataset of labeled tweets to train and test our models, with the goal of achieving high accuracy and precision in predicting depression severity levels. Our approach will involve pre-processing the data to remove noise and irrelevant information, feature extraction to identify important characteristics of the data, and training and evaluation of several machine learning models, including decision trees, random forests, and support vector machines. Ultimately, our objective is to develop a robust and accurate system that can classify depression with high accuracy and provide valuable insights into the diagnosis and treatment of this complex disorder.

## 1.1 OBJECTIVE

The objective is to achieve higher performance and shorter training times that can predict the severity of depression in tweet-length texts. Automated emotional state extraction from user text activity (e.g., posts, tweets) on social networks is the goal of emotion analysis. In earlier times, studies on emotion analysis on Twitter tended to either focus on positive or negative posts. The technique seeks to improve performance while taking less training time to predict the level of sadness in brief texts like tweets. Automated emotional state extraction from social network text activity is the aim of sentimental analysis. Early studies on emotion analysis generally used either a positive/negative bipartition or an optimistic tri-partition. Social media data seems to have the characteristics of big data on a large volume. There are multiple options to process huge amounts of information: batch processing and stream processing. The dataset of labelled tweets used in this paper has not before undergone this type of investigation because it is a brand-new dataset. Also, the majority of research on the detection of depression has concentrated on binary classification, which categorizes those who are depressed. This study's major goal is to create a system with many classes that divides depressed people into different categories according to the severity of depression. Automated emotional state extraction from social network text activity has become an increasingly popular research topic in recent years, as it has the potential to provide valuable insights into people's mental health and emotional well-being. One specific area of interest is the detection of depression in social media data, as it is a widespread and debilitating condition that can be difficult to diagnose and treat. Previous studies on emotion analysis on Twitter have typically focused on the classification of positive or negative posts. However, this approach may not be sufficient for detecting depression, as it is a complex and nuanced condition that can manifest in various ways. Therefore, the goal of this study is to develop a sentiment analysis technique that can predict the severity of depression in tweet-length texts, taking into account a wide range of emotional states. One of the challenges of

developing such a technique is achieving high performance while minimizing training time. With the increasing volume of social media data, processing large datasets can be time-consuming and resource-intensive.

The proposed technique aims to improve performance while reducing training time, using efficient algorithms and optimization techniques. Another challenge is the nature of social media data, which can be noisy, unstructured, and context-dependent. To address this challenge, the sentiment analysis technique will incorporate advanced natural language processing techniques, such as part-of-speech tagging, named entity recognition, and sentiment lexicons. These techniques can help to capture the linguistic and semantic features of the texts and extract relevant emotional information. The dataset of labelled tweets used in this study is a brand-new dataset, which has not undergone this type of investigation before. This presents an opportunity to contribute to the literature on depression detection in social media data and to evaluate the proposed technique against state-of-the-art methods. Moreover, the proposed technique aims to move beyond binary classification, which categorizes those who are depressed and those who are not. Instead, the goal is to create a system with many classes that can differentiate depressed people according to the severity of their depression, providing more personalized and targeted interventions. Overall, the proposed sentiment analysis technique has the potential to improve the accuracy and efficiency of depression detection in social media data, contributing to the development of more effective mental health screening and treatment programs. The project objective aims to address the limitations of earlier studies on emotion analysis and depression detection and to push the boundaries of sentiment analysis in the context of mental health.

## **CHAPTER – 2**

### **LITERATURE SURVEY**

#### **2.1. INTRODUCTION**

By summarizing the previously published work, a literature review serves as a tool to provide the work's context. The following references served as the basis for the creative conceptions and ideas that went into the conception of this system. This has aided in learning more about how the current systems operate and how their processes work. Since then, incorporating the benefits and drawbacks of several existing systems has assisted in changing the project.

#### **2.2 LITERATURE REVIEW**

[1] Owing to its devastating effects on a vast number of people around the world, spiteful of age, gender, or line of employment, depression has become a more common term in today's society. In light of this scenario, a human physiology-based medical diagnosis system is more required than others. Human physiology-based studies are necessary for identifying mental depression in people. Furthermore, research on the diagnosis of depression using facial reactions have been carried out in the future. This study presents a comprehensive examination of the techniques for the identification and prognosis of depression, along with an overview of the many measures, applications, and challenges faced in the field. Additionally, a review of well-known feature selection techniques is provided. The study related to relation of facial emotions of person with his mental Depression is also done here. The various researches in this field are focused which becomes more helpful for further future work related to this field

[2] The galvanic skin response, electrocardiogram, and other suitable sensors are used by the depression detector to collect an individual's physiological data and discriminate it from ordinary one. Those prompts were pre-processed to isolate the



desired elements that characterize the severity of depression in working people. The SVM and KNN algorithms are investigated to categorize this collected feature set. The result demonstrates that the optimal feature vector has a significant impact on identifying depression. It is desired to have the best feature set for getting the highest classification accuracy. The proposed methodologies are applied to the benchmark SWELL-KW dataset, and cutting-edge results are obtained.

[3] Although persistent depression is detrimental to one's emotional and physical well-being, one can fight depression by first finding it. Using information from a commercial wearable monitor, we present a strategy in this study for continuously detecting depressive episodes. Three machine-learning elements make up the method: a research lab depression detector that looks for short-term depression every two minutes; an interaction recognizer that continuously monitors user activity and provides context information; and just a context-based depression detector that uses the output of the experimental depression analyzer and the user's context to make a final conclusion every 20 minutes. Both a lab and real-world environment were used to evaluate the procedure. For a 2- class problem, the efficiency on 3 weeks of real-world data was 92%. A software for managing mental health and wellbeing now incorporates the strategy.

[4] Despite the fact that depression has become a serious health concern, modern depression detectors are difficult to use over an extended period in real life because users must either wear specialized devices or exert a lot of effort to engage with the system in order for it to be tailored to each individual. People differ substantially in their interpretation of depression and their reactions to it, necessitating adaptation. However, typical adaptation uses supervised learning techniques, necessitating reasonable sets of annotated data by each user. To get over these problems, we present a special unsupervised Depression detector that exclusively uses a smartphone and discrete HMM with MPM decisions for the interpretation of phone data. There detector is absolutely unobtrusive and suited for lifetime usage because

it doesn't require any further hardware or data labelling. Two real world datasets were used to assess its accuracy: the first used extremely brief (a few days) phone interaction records for everyone, and the second used lengthier histories. The suggested HMM-MPM in these experiments attained accuracy levels of 59 and 70%, respectively. These results are equivalent to those of fully supervised systems as described in earlier publications.

[5] In this study, a framework for identifying and analyzing depressive and anxious mental states using recorded facial cues is developed. A rigorous experimental methodology was created to induce systematic variety in emotional states (mild, relaxed, and depressed or anxious) by varying the level of external and internal depression. The study mainly focused upon non-voluntary and semi-voluntary facial cues in order to evaluate the sentiment representation more precisely. The study mainly focused upon non-voluntary and semi-voluntary facial cues in order to evaluate the sentiment representation more precisely. The heart rate was obtained using camera-based photoplethysmography. The most accurate features in each experimental phase were selected using a feature selection technique, and then classification techniques were used to differentiate between neutral and depressive or anxious states with reference to a relaxed condition. Additionally, a ranking transformation utilizing self-reports was recommended to examine the connection between facial features and the subject's reported levels of anxiety or sadness. The research revealed that certain facial cues accumulated through oral tradition, sight, and shifting heads as well as cardiac activity detected by a camera are reliable and appropriate as predictors of depression and anxiety.

[6] Face emotion recognition is becoming a common practice and a fascinating area of affective computing, particularly for computer vision-based healthcare applications. Many situations call for diverse facial expressions depending on the person and the moment. Facial expressions play a crucial part in human-machine interfaces and are used by computers to detect emotions automatically. The ability

of computers to recognize people from their facial expressions is still difficult. The study that is being presented suggests emergence-based Eigen-face approaches. PCA (Principal Component Analysis) allows us to extract all pertinent data from frames with recognized human faces. We are aware of how emotions are being expressed through facial expressions. We use PCA to reduce computations' dimensionality. In this process, faces are initially detected, followed by the extraction of features, the reduction of dimensionality by PCA, and finally the classification of emotions via the Euclidean separation metric. Finally, we minimize emotions by removing unnecessary frames using time-dependent processes (Patthe and Anil, Temporal dynamics of continuous face emotion recognition mechanisms, 2017). The eigenvectors were determined using the training set of images, thereby generating the face spaces. We use PCA to compress various orientations as well as the pertinent frame scale. We utilized a database for PCA that contains certain frames needed for training. The suggestion is tested using the remaining frames. Training frames for feelings such as annoyance, contempt, happiness, neutrality, and surprise were used. In our experiments, we used the Indian Face Database. A total of fifty frames are utilized for testing, and thirty frames are used to train the system from this database.

[7] Depression is a physical condition of the mind that developed in response to a hard or stressful event. External variables that contribute to depression are known as depressive influences. Extended exposure to numerous depressive effects at once may have a harmful effect on the mental and physical well-being of an individual, which may further result in chronic health problems. The only way to prevent issues caused by depression is to recognize them while they are still in their infancy, which is only possible by vigilantly tracking depression. One can track their own degrees of depression thanks to the continuous and real-time data collection provided by wearable computing. Subject of this paper's thorough review is the application of wearable sensors and machine learning approaches to identify depression. This study examines the methods for detecting depression that are used in accordance with

wearable sensors, electrocardiograms, electroencephalograms, and PPG, as well as in various settings, including when driving, studying, and working. Future research studies are anticipated to follow the same methods, outcomes, benefits, limitations, and difficulties for each study. At the conclusion, a wearable sensor-based multimodal Depression detection system has also been proposed.

## **CHAPTER-3**

### **SYSTEM ANALYSIS**

#### **3.1. EXISTING SYSTEM**

Depression, in general terms, is defined as any kind of disturbance in physiological homeostasis. Thus, PS is the homeostatic alteration caused by psychological factors which may include various social and emotional Depressions. The concept of PS has slowly evolved from a neurobiological to a neuro-physiological basis. This development is evident from the methods of qualitative and quantitative assessment of Depression, which have improvised from the classical questionnaire-based evaluation of Depression to current molecular screening methods. Initially, PS was considered purely from a psychiatry viewpoint. Consequently, various questionnaires were used and are still used for the psychological assessment of an individual. These questionnaires are designed based on various factors like daily hassles, happiness scale, perception of the present and the future by an individual (e.g., optimism or pessimism), personality traits, depressive life events, and so on. Although these questionnaires, with certain scoring methods, can assess the degree of Depression involved, but there are several drawbacks, such as, The subjective bias of an individual towards the assessment of his/her psychological state, The tendency of an individual to maintain secrecy regarding personal matters while avoiding any disclosure even when confidentiality is maintained, The limitation of questionnaires in terms of their objectivity thus restricted to investigate an individual without prior bias. All these factors, therefore, pose an upper limit to questionnaire-based evaluation of an individual for PS and raise serious queries regarding validation of the estimation. Thus, for a more comprehensive and robust evaluation of PS, hormonal assays came into picture in which levels of epinephrine, norepinephrine, and cortisol are checked to differentiate various Depression conditions. In addition, other physiological studies which

include estimation of various clinical parameters, such as elevated blood pressure, rise in body temperature and changes in body weight, have also been used to measure PS response. However, these studies failed to estimate and quantify degree of PS in different scenarios. These problems, associated with hormonal assays and physiological readings, have been dealt with by expanding the studies of PS quantification to further downstream products of depression response rather than restricting to primary hormonal responses as well as using various tissues to acquire the final read-outs.

### **3.1.1. DRAWBACKS OF EXISTING SYSTEM**

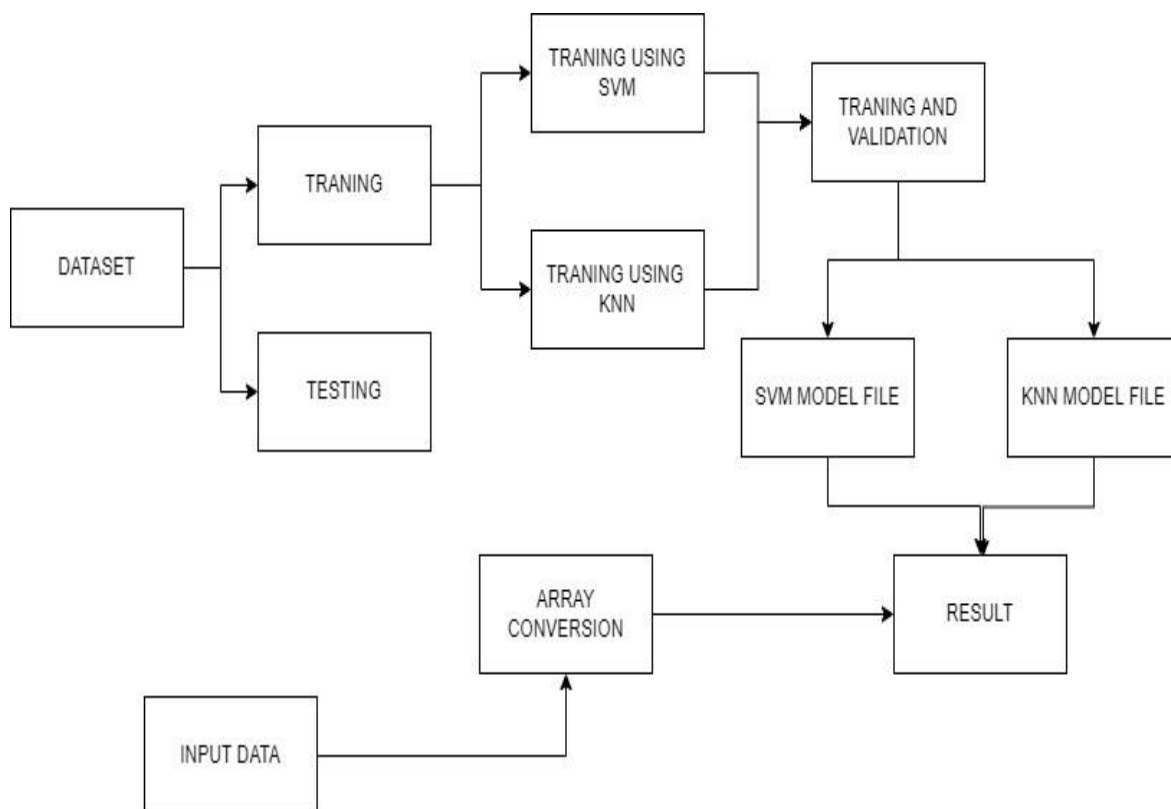
While existing systems for detecting depression from tweets have shown promising results, they also have some drawbacks. Here are a few examples:

- **Limited data availability:** Most existing systems for detecting depression from tweets have been trained on a relatively small and biased data set, which may not be representative of the diverse population. This can lead to the development of models that are not sensitive to cultural and linguistic differences, which can affect their accuracy and reliability.
- **Inability to capture context:** Existing systems for detecting depression from tweets often focus on individual words or phrases, rather than considering the context in which they are used. This can lead to misclassification of tweets, as some words or phrases may have different meanings depending on the context in which they are used.
- **Privacy concerns:** The use of social media data for detecting depression raises privacy concerns, as it involves the collection and analysis of personal information.

Overall, these drawbacks highlight the need for further research and development in the area of depression detection from tweets, with a focus on improving the accuracy and reliability of existing systems while addressing the ethical and privacy concerns associated with their use.

### 3.2. PROPOSED SYSTEM

The technique seeks to improve performance while taking less training time to predict the level of sadness in brief texts like tweets. Automated emotional state extraction from social network text activity is the aim of sentimental analysis. Early studies on emotion analysis generally used either a positive/negative bipartition or an optimistic tri-partition. Social media data seems to have the characteristics of big data on a large volume. There are multiple options to process huge amounts of information: batch processing and stream processing. The dataset of labelled tweets used in this paper has not before undergone this type of investigation because it is a brand-new dataset. Also, the majority of research on the detection of depression has concentrated on binary classification, which categorizes those who are depressed. This study's major goal is to create a system with many classes that divides depressed people into different categories according to the severity of depression.



**Fig.1. BLOCK DIAGRAM**

The data is being collected in the form of labelled tweets that portrays the depressed and not depressed tweets of peoples, and the collected data are being trained with these conditions. A model file has been developed for acquiring the input text. An SVM and KNN algorithm are acquired to obtain the result of individual's depression level.



## **CHAPTER-4**

### **SYSTEM SPECIFICATION**

#### **4.1. INTRODUCTION**

A clear view of the hardware and software requirements of the project is crucial to understand its commissioning and working. The chapter takes a deep analysis of the various software requirements for the proposed system.

#### **4.2. HARDWARE SPECIFICATIONS**

This section gives the details and specification of the hardware on which the system is expected to work.

Processor : AMD Ryzen  
RAM : 8 GB SD RAM  
Hard disk : 50GB SSD

#### **4.3. SOFTWARE SPECIFICATIONS**

This section gives the details of the software that are used for the development.

- Python 3.7
- Google Colab

#### **4.4. SOFTWARE DESCRIPTION**

##### **4.4.1. PYTHON PROGRAMMING LANGUAGE**

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

## FEATURES

- Easy-to-learn - Python includes a small number of keywords, precise structure, and well-defined syntax. This allows the student to learn the language faster
- Easy to read – Python code is clearly defined and visible to the naked eye.
- Easy-to-maintain - Python source code is easy to maintain.
- Standard General Library - Python's bulk library is very portable and shortcut compatible with UNIX, Windows, and Macintosh.
- Interaction mode - Python supports an interaction mode that allows interaction testing and correction of captions errors.
- Portable - Python works on a variety of computer systems and has the same user interface for all. Extensible - Low-level modules can be added to the Python interpreter. These modules allow system developers to improve the efficiency of their tools either by installing or customizing them.
- Details - All major commercial information is provided by Python ways of meeting.
- GUI Programming - Python assists with the creation and installation
- of a user interface for images of various program phones, libraries
- and applications, including Windows MFC, Macintosh, and Unix's Window.
- Scalable - Major projects benefit from Python building and support, the Shell writing is not.

### 4.4.2. PYTHON 3.7

Python is an interpreter, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace.

Python is an easy to learn, powerful programming language. It has efficient high-

level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many a reason most platforms and may be freely distributed. The same site also contains distributions of and pointers to many free third-party Python modules, programs and tools, and additional documentation. The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications. This tutorial introduces the reader informally to the basic concepts and features of the Python language and system. It helps to have a Python interpreter handy for hands-on experience, but all examples are self-contained, so the tutorial can be read off-line as well. For a description of standard objects and modules, see [library-index](#). [Reference-index](#) gives a more formal definition of the language. To write extensions in C or C++, read [extending-index](#) and [c-api-index](#). There are also several books covering Python in depth.

This tutorial does not attempt to be comprehensive and cover every single feature, or even every commonly used feature. Instead, it introduces many of Python's most notes worthy features, and will give you a good idea of the language's flavor and style. After reading it, you will be able to read and write Python modules and programs, and you will be ready to learn more about the various Python library modules described in [library-index](#). If you do much work on computers, eventually you find that there's some tasks you'd like to automate. For example, you may wish to perform a search-and-replace over a large number of text files, or rename and rearrange a bunch of photo files in a complicated way. Perhaps you'd like to write a small custom database, or a specialized

GUI application or a simple game. If you're a professional software developer, you may have to work with several C/C++/Java libraries but find the usual write/compile/test/re-compile cycle is too slow. Perhaps you're writing a test suite

for such a library and find writing the testing code a tedious task. Or maybe you've written a program that could use an extension language, and you don't want to design and implement a whole new language for your application.

#### **4.4.3. GOOGLE COLAB:**

Google Colab was developed by Google to provide free access to Graphics Processing Unit and Tensor Processing Unit to anyone who needs them to build a machine learning or deep learning model. Google Colab can be defined as an improved version of Jupyter Notebook. One of the key benefits of Google Colab is that it provides access to powerful hardware resources, including GPUs and TPUs, which are essential for training deep learning models that require a large amount of computational power. The platform also allows users to access and use various Machine Learning and Deep Learning libraries such as TensorFlow, PyTorch, and Keras, making it a popular choice among data scientists and machine learning enthusiasts.

#### **4.4.4 MODULES & PACKAGES**

- **PANDAS**

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals. Its name is a play on the phrase "Python data analysis" itself. Wes McKinney started building what would become pandas at AQR Capital while he was a researcher there from 2007 to 2010.

- **FEATURES:**

- Many inbuilt methods available for fast data manipulation made possible with vectorization.
- Data Frame object for multivariate data manipulation with integrated indexing.
- Series object for univariate data manipulation with integrated indexing
- Tools for reading and writing data between in-memory data structures and different file formats.
- Data alignment and integrated handling of missing data.
- Reshaping and pivoting of data sets.
- Label-based slicing, fancy indexing, and sub setting of large data sets.
- Data structure column insertion and deletion.
- Group by engine allowing split-apply-combine operations on data sets.
- Data set merging and joining.

- **NUMPY**

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

- **SCIKIT-LEARN**

Scikit-learn is probably the most useful library for machine learning in Python. TheSklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, and clustering and dimensionality reduction.

- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

## • **MATPLOTLIB**

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

### **FEATURES:**

- Create publication quality plots.
- Make interactive Figures that can zoom, pan, and update.
- Customize visual style and layout.
- Export to many file formats.
- Embed in Jupyter Lab and Graphical User Interfaces.
- Use a rich array of third-party packages built on Matplotlib.

## • **SEABORN**

Seaborn is a library for making statistical graphics in Python. It is built on top of matplotlib and closely integrated with pandas data structures.

Here is some of the functionality that seaborn offers:

- A dataset-oriented API for examining relationships between multiple variables
- Specialized support for using categorical variables to show observations or aggregate statistics
- Options for visualizing univariate or bivariate distributions and for comparing them between subsets of data

- Automatic estimation and plotting of linear regression models for different kinds of dependent variables
- Convenient views onto the overall structure of complex datasets
- High-level abstractions for structuring multi-plot grids that let you easily build complex visualizations.

## **CHAPTER-5**

### **SYSTEM OVERVIEW**

#### **5.1. MODULE DESCRIPTION**

##### **5.1.1 DATASET**

Collect data from individuals who have been clinically diagnosed with depression and individuals who have not been diagnosed with depression. The data should include input features such as age, gender, family history, lifestyle factors, and symptoms of depression. The dataset for this study collected from Kaggle, and it contains 20000 tweets from a variety of people. -e users name, tweet date, and depression impact tweet were all included in the data collection. The user's tweets were preprocessed before the experiment to remove stop words, special characters, and symbols that might cause the polarity of the tweets to worsen. There are 2 fields in the dataset:

Text: the tweet data.

Label: the polarity of tweets (0 = negative and  
1 = positive).

In the imbalanced dataset, first the number of instances with positive and negative tweets is checked. This will de-grade the overall predictive performance of the model.

##### **5.1.2 SVM CLASSIFIER**

SVM algorithm approaches classification and regression issues using an SVM classifier and an SVM regressor, respectively. The SVM classifier, on the other hand, is the cornerstone of the support vector machine idea and generally speaking, the best approach for dealing with classification issues. Being a linear algorithm at



its core can be imagined almost like a Linear or Logistic Regression. For example, an SVM classifier creates a line (plane or hyper-plane, depending upon the dimensionality of the data) in an N-dimensional space to classify data points that belong to two separate classes. It is also noteworthy that the original SVM classifier had this objective and was originally designed to solve binary classification problems, however unlike, say, linear regression that uses the concept of line of best fit, which is the predictive line that gives the minimum Sum of Squared Error (if using OLS Regression), or Logistic Regression that uses Maximum Likelihood Estimation to find the best fitting sigmoid curve, Support Vector Machines uses the concept of Margins to come up with predictions.

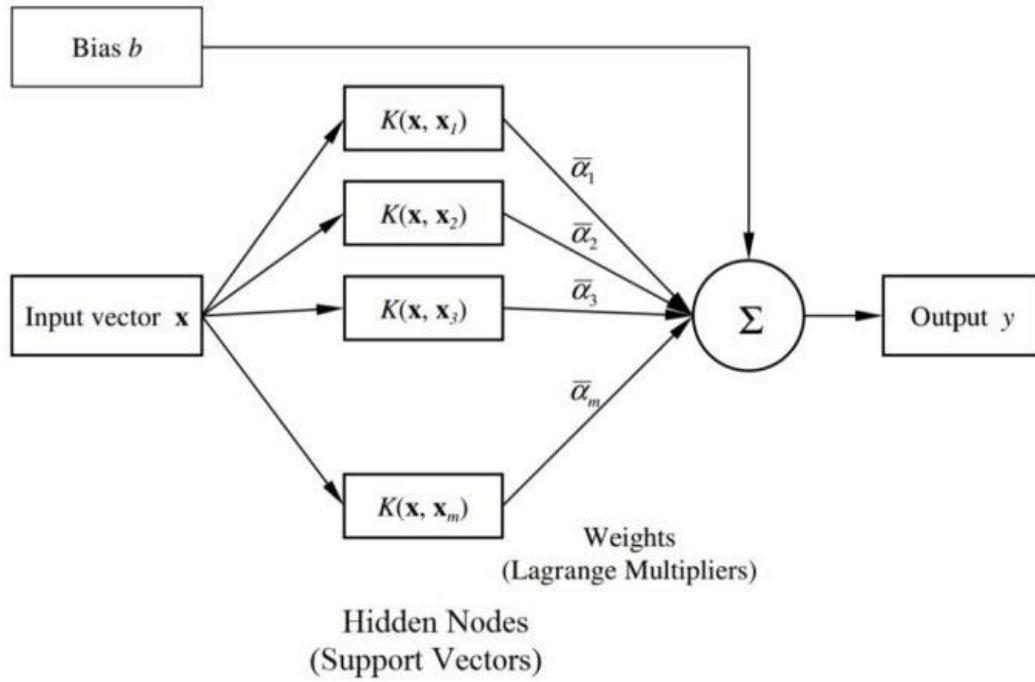
One of the most widely used supervised learning methods is the SVM because it can be applied to both categorization and regression problems. A form of machine learning algorithm known as a support vector machine modifies features by using kernel techniques. In order to make it simpler to distinguish between the classes after the transformation, kernel functions translate the data into a different, sometimes higher dimensional space. In machine learning, it is frequently used to address classification problems: A support vector machine is a model that uses the classification algorithm for two-group classification problems. An SVM is a fast and dependable algorithm. It consists of a line that separates two data objects, known as the decision boundary is acts as the main separation axis equation of the separation axis is

$$Y = mx + c$$

where m stands for the slope.

Now, the hyperplane equation separating the data objects are

H:  $wt.(x)+b = 0$ , where b stands for the bias term.



**Fig 2. Flow Diagram of SVM**

## WORKING OF SVM ALGORITHM

SVM algorithm predicts the classes. One of the classes is identified as 1 while the other is identified as 0, all machine learning algorithms convert the business problem into a mathematical equation involving unknowns. These unknowns are then found by converting the problem into an optimization problem. As optimization problems always aim at maximizing or minimizing something while looking and tweaking for the unknowns, in the case of the SVM classifier, a loss function known as the hinge loss function is used and tweaked to find the maximum margin. For ease of understanding, this loss function can also be called a cost function whose cost is 0 when no class is incorrectly predicted. However, if this is not the case, then error/loss is calculated. The problem with the current scenario is that there is a trade-off between maximizing margin and the loss generated if the margin is maximized to a very large extent. To bring these concepts in theory, a regularization parameter

is added. As is the case with most optimization problems, weights are optimized by calculating the gradients using advanced mathematical concepts of calculus viz. partial derivatives. The gradients are updated only by using the regularization parameter when there is no error in the classification while the loss function is also used when misclassification happens. The gradients are updated only by using the regularization parameter when there is no error in the classification, while the loss function is also used when misclassification happens.

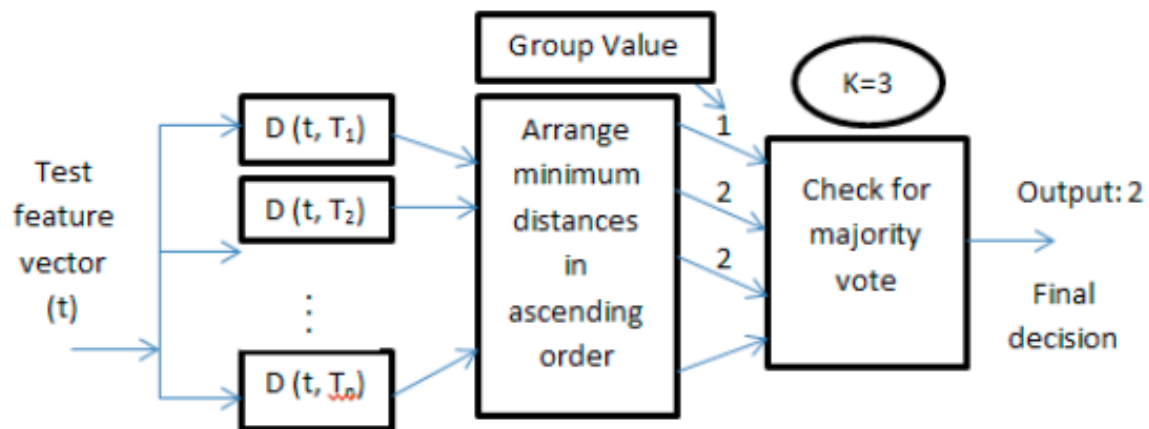
### **5.1.3. KNN CLASSIFIER**

KNN models were developed for image classification problems, where the model learns an internal representation of a two-dimensional input, in a process referred to as feature learning. This same process can be harnessed on one-dimensional sequences of data, such as in the case of acceleration and gyroscopic data for human activity recognition. The model learns to extract features from sequences of observations and how to map the internal features to different activity types. The benefit of using KNN for sequence classification is that they can learn from the raw time series data directly, and in turn do not require domain expertise to manually engineer input features. The model can learn an internal representation of the time series data and ideally achieve comparable performance to models fit on a version of the dataset with engineered features.

This section is divided into 3 parts; they are:

1. Load Data
2. Fit and Evaluate Model
3. Summarize Results

K-nearest neighbours (KNN) algorithm is a type of supervised ML algorithm which can be used for both classification as well as regression predictive problems. However, it is mainly used for classification predictive problems in industry.



**Fig-3 Flow Diagram of KNN**

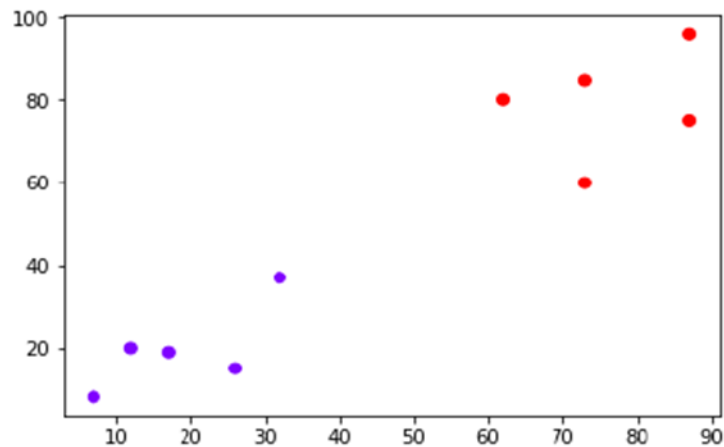
The following two properties would define KNN well –

- Lazy learning algorithm – KNN is a lazy learning algorithm because it does not have a specialized training phase and uses all the data for training while classification.
- Non-parametric learning algorithm – KNN is also a non-parametric learning algorithm because it does not assume anything about the underlying data.

## WORKING OF KNN ALGORITHM

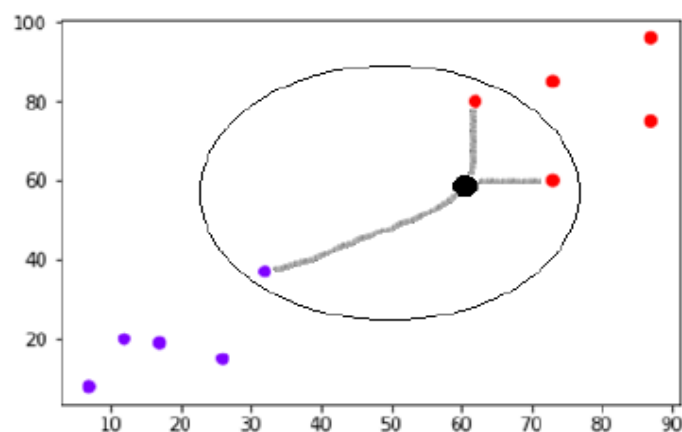
K-nearest neighbours (KNN) algorithm uses ‘feature similarity’ to predict the values of new data points which further means that the new data point will be assigned a value based on how closely it matches the points in the training set. We can understand its working with the help of following steps: For implementing any algorithm, we need dataset. So during the first step of KNN, we must load the training as well as test data. Next, we need to choose the value of K i.e. the nearest data points. K can be any integer. For each point in the test data do the following, Calculate the distance between test data and each row of training data with the help of any of the method namely: Euclidean, Manhattan or Hamming distance. The most used method to calculate distance is Euclidean. Now, based on the distance value,

sort them in ascending order. Next, it will choose the top K rows from the sorted array. Now, it will assign a class to the test point based on most frequent class of these rows. Suppose we have a dataset which can be plotted as follows:



**Fig 4 KNN Dataset Plotted**

Now, we need to classify new data point with black dot (at point 60,60) into blue or red class. We are assuming  $K = 3$  i.e., it would find three nearest data points. It is shown in the next diagram –



**Fig 5 Classify new data point**

We can see in the above diagram the three nearest neighbors of the data point with

black dot. Among those three, two of them lies in red class hence the black dot will also be assigned in red class.

#### **5.1.4. PREPROCESSING**

Preprocessing the collected data is an important step in preparing it for machine learning. This may involve cleaning the data, handling missing values, and normalizing the features. It is also important to carefully consider which features to include in the models, as including irrelevant or redundant features can negatively impact their performance. The pre- processing methods we used were histogram equalization, performance evaluation metrics like accuracy, F1-score sensitivity, precision, and Matthews Correlation Coefficient , and image resizing.

#### **5.1.5. MODEL SELECTION**

While SVM and KNN are popular machine learning algorithms that can be used for depression detection, other algorithms such as decision trees, random forests, and neural networks may also be effective. It is important to carefully evaluate the performance of each model and choose the one that performs the best on the testing set.

#### **5.1.6. PERFORMANCE EVALUATION METRICS**

For each model, The models were assessed for precision, sensitivity, F1-score, accuracy, and correctness with respect to MCC. The following equations emphasize these metrics:

$$Precision = (TP + TN)/((TP + FP + TN + FN) ) \quad (2)$$

where TP stands for True Positive, FP for False Positive, TN for True Negative, and FN for False Negative. The TP rate (TPR) is calculated to know the percentage of positive data that was expected to be positive. The term sensitivity also refers to the true- positive rate. The degree to which repeated measurements under constant circumstances yield the same findings is what is meant by the term precision.

$$TPR = ((TP + FN))/TP \quad (3)$$

The F1 score, a machine learning evaluation metric, rates a model's precision. It incorporates a model's recall and precision ratings. The accuracy metric shows how many times a model properly predicted over the entire dataset. This measure can only be trusted if the dataset is class-balanced, meaning that each class contains an equal number of samples.

$$F1 \text{ Score} = 2(Precision + Recall)/(Precision + Recall) \quad (4)$$

Following feature extraction testing, the set of photos is processed, the test image is classed, and the accuracy is plotted.

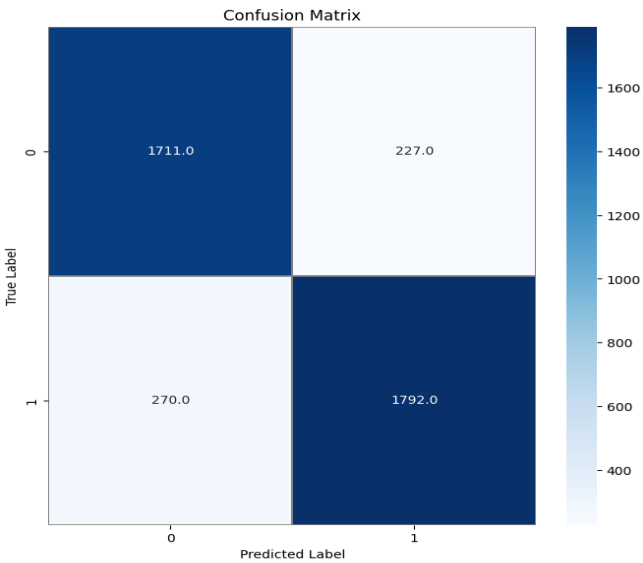
Utilizing accuracy, the model's performance is assessed. As used in other studies, it is defined as the proportion of correctly identified cases divided by the total number of instances in the dataset. When evaluating the performance of the models, it is important to consider multiple metrics such as accuracy, precision, recall, and F1-score. These metrics can help provide a more comprehensive evaluation of the models' performance, particularly when dealing with imbalanced datasets.

### **5.1.7. GENERALIZATION**

It is important to ensure that the models can generalize to new, unseen data. This can be achieved by using techniques such as cross-validation, regularization, and data augmentation. It is also important to test the models on data from different sources to ensure that they can generalize to different populations. When building machine learning models for classifying depression using social media tweets, it is essential to ensure that the models are capable of generalizing to new, unseen data. This is because the ultimate goal of the model is to be able to accurately classify depression in any new social media data that comes in, not just the data that was used to train the model. To ensure that the models can generalize well, techniques such as cross-validation, regularization, and data augmentation can be used. Cross-validation involves dividing the available data into multiple subsets, with each subset used to

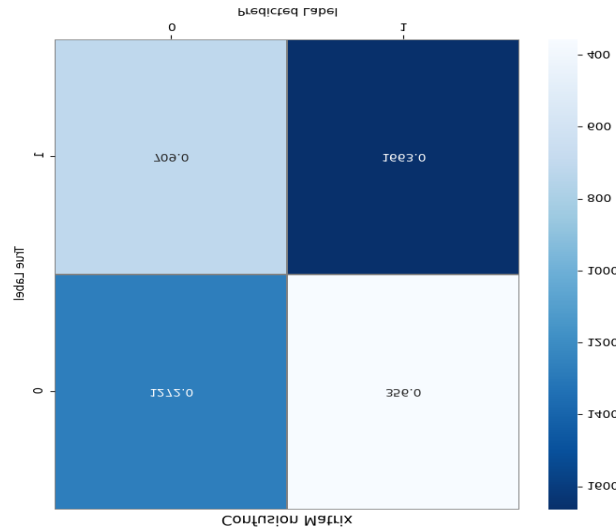
train and test the model at different times. This helps to ensure that the model is not overfitting to the training data and is generalizing well to new, unseen data. Regularization is another technique that can help to prevent overfitting of the model to the training data. A penalty term is added to the model's cost function to discourage the model from learning overly complex relationships in the data. Data augmentation is the process of generating new data samples by applying transformations to existing data. This can help to increase the amount of available training data and improve the model's ability to generalize to new, unseen data.

**5.1.8. CONFUSION MATRIX**



**Fig.6. Confusion Matrix for SVM**





**Fig.7. Confusion Matrix for KNN**

Fig. 6 and Fig 7 gives the confusion matrix of SVM and KNN classifier respectively, which used to assess how well the classification models work given a particular set of test data. Only when the real test data values are known can it be decided. The matrix itself is simple to understand, but the associated terms might be. Since it displays the mistakes in the model performance as a matrix, it is also referred to as an error matrix.

#### 5.1.9. CLASSIFIED RESULT

TABLE 1. SVM ACCURACY

	Precision	Recall	f1-score	Support
<b>Class 1</b>	0.86	0.88	0.87	1938
<b>Class 2</b>	0.89	0.87	0.88	2062
<b>Accuracy</b>			0.88	4000
<b>Macro Avg</b>	0.88	0.88	0.88	4000
<b>Weighted Avg</b>	0.88	0.88	0.88	4000

This is a classification report that summarizes the performance of a binary classification model on a dataset consisting of 4000 samples. The two classes are labelled as Class 1 and Class 2, with 1938 and 2062 samples, respectively. The precision of Class 1 is 0.86, which means that when the model predicted a sample to be Class 1, it was correct 86% of the time. The recall of Class 1 is 0.88, which means that the model correctly identified 88% of all samples that belonged to Class 1. The F1-score of Class 1 is 0.87, which is the harmonic mean of precision and recall. Similarly, for Class 2, the precision is 0.89, which means that when the model predicted a sample to be Class 2, it was correct 89% of the time. The recall of Class 2 is 0.87, which means that the model correctly identified 87% of all samples that belonged to Class 2. The F1-score of Class 2 is 0.88. The accuracy of the model is 0.88, which means that it correctly classified 88% of all samples. The macro-average of precision, recall, and F1-score is also 0.88, which indicates that the model performs equally well on both classes. The weighted-average of precision, recall, and F1-score is also 0.88, which indicates that the model is not biased towards either class, and the performance is evenly distributed across both classes. In summary, this classification model shows good performance with an overall accuracy of 0.88 and balanced precision, recall and F1-score for both classes.

TABLE 1. KNN ACCURACY

	<b>precision</b>	<b>Recall</b>	<b>f1-score</b>	<b>Support</b>
<b>Class 1</b>	0.64	0.78	0.70	1628
<b>Class 2</b>	0.82	0.70	0.76	2372
<b>Accuracy</b>			0.73	4000
<b>macro avg</b>	0.73	0.74	0.73	4000
<b>weighted avg</b>	0.75	0.73	0.74	4000

The precision for Class 1 is 0.64, which means that when the model predicts an instance as Class 1, it is correct 64% of the time. The recall for Class 1 is 0.78, which means that of all the instances that belong to Class 1, the model correctly identifies

78% of them. The F1-score for Class 1 is 0.70, which is the harmonic mean of precision and recall. The precision for Class 2 is 0.82, which means that when the model predicts an instance as Class 2, it is correct 82% of the time. The recall for Class 2 is 0.70, which means that of all the instances that belong to Class 2, the model correctly identifies 70% of them. The F1-score for Class 2 is 0.76. The overall accuracy of the model is 0.73, which means that the model correctly predicts the class of 73% of the instances in the dataset. The macro average of precision, recall, and F1-score is the average of the values for both classes. In this case, it is 0.73, which is the same as the overall accuracy. The weighted average of precision, recall, and F1-score is calculated based on the number of instances in each class. In this case, the weighted average is 0.75 for precision and 0.73 for both recall and F1-score. This indicates that the model performs slightly better on Class 2, which has more instances in the dataset.

## **CHAPTER 6**

### **SYSTEM IMPLEMENTATION**

#### **6.1. SYSTEM IMPLEMENTATION**

The research described here presents a machine-learning data classification system that uses the SVM and KNN algorithms to classify the data. We theoretically analyzed and measured each person's psychological depression level, and then used the dataset to verify our findings experimentally. Significant biomarkers that are related to psychological depression have also been discovered, allowing for the prediction of depression status in unidentified people. Collecting information on both individuals with a depression diagnosis that has been supported by clinical evidence and people who do not have a diagnosis of depression. Age, gender, family background, lifestyle factors, and depressive symptoms should all be input features in the data. The gathered data is cleaned, missing values are removed, and the features are normalized as part of the preprocessing. Separate the training and test groups from the preprocessed data. The SVM and KNN models will be trained using the training set, and their efficacy will be assessed using the testing set. By utilizing the training set to build an SVM model. Based on the input features, the SVM model will learn to categorize whether an individual is depressed or not. Utilize the training data to train a KNN model. Based on the input features, the KNN model will learn to categorize whether an individual is depressed or not. Applying the testing set, assess how well the SVM and KNN models work. Pick the model that achieved the best on the testing group. Apply the selected paradigm to identify depression. It is crucial to remember that detecting depression is a challenging job, and that using SVM and KNN alone might not produce reliable results. To enhance the performance of the models, it is advised to combine these algorithms with other methods like feature selection, data enrichment, and ensemble learning.

The entire system was developed using the Python and MachineLearning.

The process consists of three steps

1. Data Preparation
2. Model Comparison
3. Differentiation

#### **6.1.1. DATA PREPARATION**

Data preparation is an important step in preparing data for machine learning. This may involve cleaning the data, handling missing values, and normalizing the features. It is also important to carefully consider which features to include in the models, as including irrelevant or redundant features can negatively impact their performance.

#### **6.1.2. MODEL COMPARISON**

While SVM and KNN are popular machine learning algorithms that can be used for depression detection, other algorithms such as decision trees, random forests, and neural networks may also be effective. It is important to carefully evaluate the performance of each model and choose the one that performs the best on the testing set.

#### **6.1.3. DIFFERENTIATION**

It is important to ensure that the models can generalize to new, unseen data. This can be achieved by using techniques such as cross-validation, regularization, and data augmentation. It is also important to test the models on data from different sources to ensure that they can generalize to different populations.

## CHAPTER-7

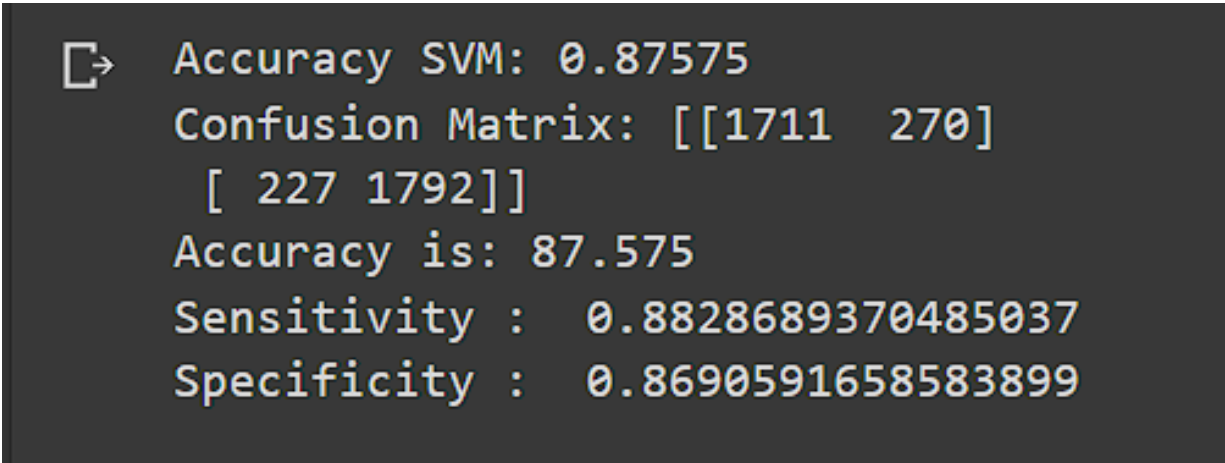
### RESULT AND OUTPUT

#### 7.1. EXPERIMENTAL RESULT

The following describes the accuracy rate and confusion matrix obtained by SVM and KNN.

##### 7.1.1 ACCURACY RATE OF SVM

The accuracy rate and confusion matrix of the SVM is practically analysed and then the fact is concluded as the application provide an accuracy of more than 88%.

A screenshot of a terminal window with a dark background and light-colored text. It displays the following output: Accuracy SVM: 0.87575, Confusion Matrix: [[1711 270] [ 227 1792]], Accuracy is: 87.575, Sensitivity : 0.8828689370485037, and Specificity : 0.8690591658583899.

```
➜ Accuracy SVM: 0.87575
Confusion Matrix: [[1711  270]
 [ 227 1792]]
Accuracy is: 87.575
Sensitivity : 0.8828689370485037
Specificity : 0.8690591658583899
```

**Fig.8 Accuracy Rate of SVM**

### 7.1.2 ACCURACY RATE OF KNN

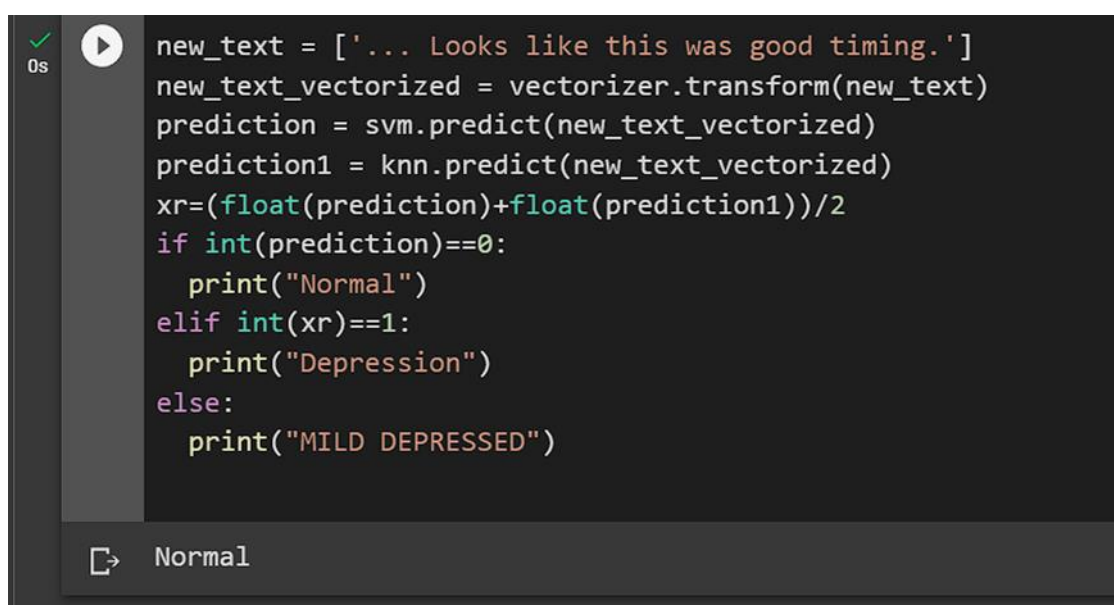
The accuracy rate and confusion matrix of the KNN is practically analysed and then the fact is concluded as the application provide an accuracy of more than 76%

```
Accuracy KNN: 0.73375
Confusion Matrix: [[1272  709]
 [ 356 1663]]
Accuracy is: 73.375
Sensitivity : 0.7813267813267813
Specificity : 0.7010961214165261
```

**Fig 9. Accuracy Rate of KNN**

### 7.2 OUTPUT

A Normal text is given as input and output is obtained as Normal.

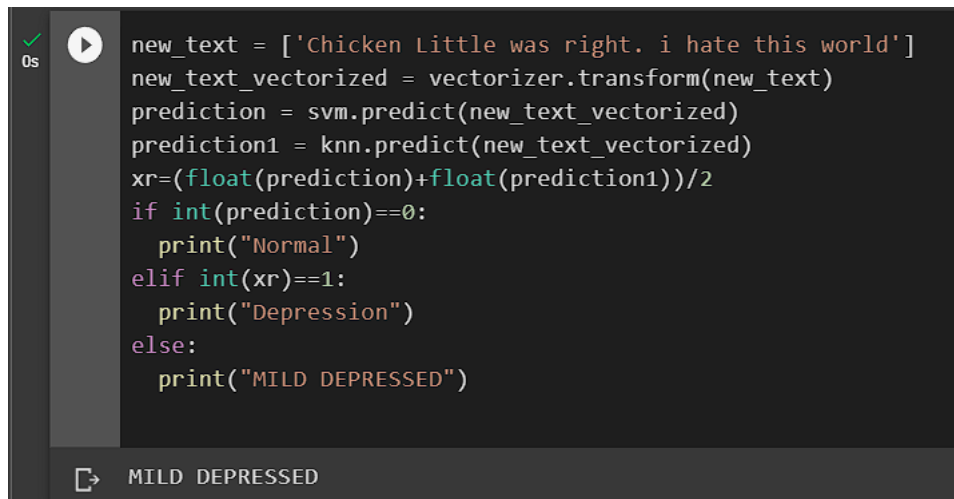


```
new_text = ['... Looks like this was good timing.']
new_text_vectorized = vectorizer.transform(new_text)
prediction = svm.predict(new_text_vectorized)
prediction1 = knn.predict(new_text_vectorized)
xr=(float(prediction)+float(prediction1))/2
if int(prediction)==0:
    print("Normal")
elif int(xr)==1:
    print("Depression")
else:
    print("MILD DEPRESSED")
```

Normal

**Fig 10 Output of Normal Prediction**

A Mild depressed text is given as input and output is obtained as Mild depressed.

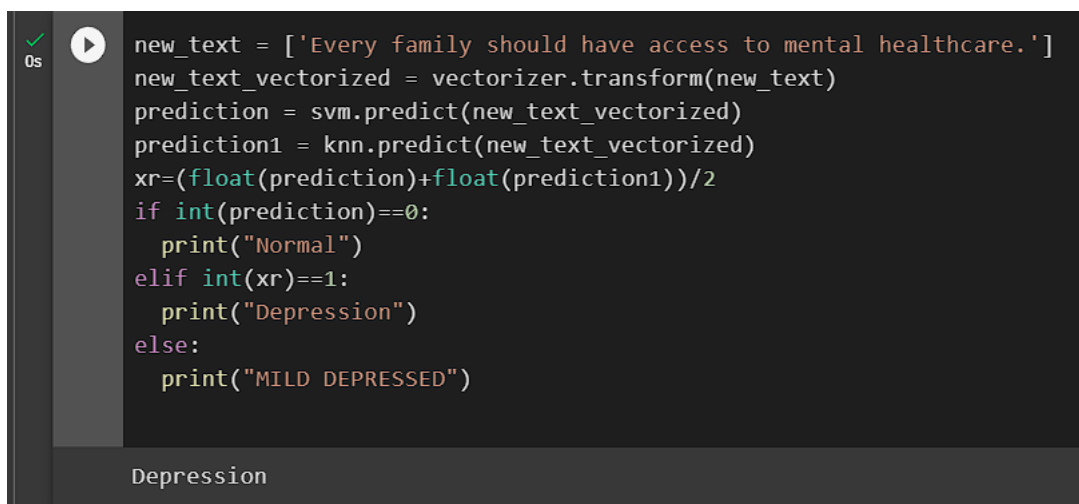


```
✓ 0s ▶ new_text = ['Chicken Little was right. i hate this world']
new_text_vectorized = vectorizer.transform(new_text)
prediction = svm.predict(new_text_vectorized)
prediction1 = knn.predict(new_text_vectorized)
xr=(float(prediction)+float(prediction1))/2
if int(prediction)==0:
    print("Normal")
elif int(xr)==1:
    print("Depression")
else:
    print("MILD DEPRESSED")
```

➤ MILD DEPRESSED

**Fig 11 Output of MILD DEPRESSED Prediction**

A Depressed text is given as input and output is obtained as Depressed.



```
✓ 0s ▶ new_text = ['Every family should have access to mental healthcare.']
new_text_vectorized = vectorizer.transform(new_text)
prediction = svm.predict(new_text_vectorized)
prediction1 = knn.predict(new_text_vectorized)
xr=(float(prediction)+float(prediction1))/2
if int(prediction)==0:
    print("Normal")
elif int(xr)==1:
    print("Depression")
else:
    print("MILD DEPRESSED")
```

Depression

**Fig 12 Output of DEPRESSION Prediction**



## **CHAPTER-8**

### **CONCLUSION AND FUTURE WORK**

#### **8.1. CONCLUSION**

In conclusion, depression detection using SVM and KNN is a classification problem that can be addressed by collecting data, preprocessing it, training SVM and KNN models, and evaluating their performance. The proposed system can greatly enhance the management of mental health by predicting depression in modern living using social media tweets and SVM and KNN algorithms. This technique showed us how important feature extraction and data preparation are to the model's performance. For machine learning models to be accurate and reliable, social media data must be cleaned and preprocessed. While SVM and KNN are popular machine learning algorithms that can be used for this task, it is important to note that depression detection is a complex task that may require the use of other techniques to improve the performance of the models. Therefore, it is important to carefully design the data collection and preprocessing steps and to consider using other advanced techniques to accurately detect depression.

#### **8.2. FUTURE WORK**

Social media sites have developed into a valuable resource for gathering data on people's mental states as mental health problems become more common. Large datasets of tweets can be analyzed and categorized into groups like depression, anxiety, and stress using machine learning and natural language processing methods. Future research may look into using multimodal data to classify depression using machine learning. By adjusting machine learning models for depression categorization to the profiles and tastes of particular users, customized screening and intervention can improve the precision and effectiveness of screening and intervention. Future research can look into the clinical applications of machine learning algorithms for identifying depression on social media. Benefit of classifying depression from Twitter tweets is the ability to provide personalized

support or other resources to individuals through a platform by providing counselling to the persons who are struggling with mental health issues.

## **APPENDIX**

### **A.1. ABOUT THE APPLICATION**

The research described here presents a machine-learning data classification system that uses the SVM and KNN algorithms to classify the data. We theoretically analyzed and measured each person's psychological depression level, and then used the dataset to verify our findings experimentally. Significant biomarkers that are related to psychological depression have also been discovered, allowing for the prediction of depression status in unidentified people. Collecting information on both individuals with a depression diagnosis that has been supported by clinical evidence and people who do not have a diagnosis of depression. Age, gender, family background, lifestyle factors, and depressive symptoms should all be input features in the data. The gathered data is cleaned, missing values are removed, and the features are normalized as part of the preprocessing. Separate the training and test groups from the preprocessed data. The SVM and KNN models will be trained using the training set, and their efficacy will be assessed using the testing set. By utilizing the training set to build an SVM model. Based on the input features, the SVM model will learn to categorize whether an individual is depressed or not. Utilize the training data to train a KNN model. Based on the input features, the KNN model will learn to categorize whether an individual is depressed or not. Applying the testing set, assess how well the SVM and KNN models work. Pick the model that achieved the best on the testing group. Apply the selected paradigm to identify depression. It is crucial to remember that detecting depression is a challenging job, and that using SVM and KNN alone might not produce reliable results.

## **A.2. DEVELOPERS**

**BIPIN V K**                      bipinvk47@gmail.com

**CRIS CUMINS P**                criscumins303@gmail.com

**WILSON A**                      wilsonraja.ra@gmail.com

## **B. SOURCE CODE**

### **B.1. PYTHON CODING**

#### **SVM training and testing code:**

```
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, confusion_matrix
data = pd.read_csv('dataset.csv')
X = data['message']
y = data['label']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
vectorizer = CountVectorizer()
X_train_count = vectorizer.fit_transform(X_train)
X_test_count = vectorizer.transform(X_test)
svm = SVC(kernel='linear')
svm.fit(X_train_count, y_train)
y_pred = svm.predict(X_test_count)
accuracy = accuracy_score(y_test, y_pred)
confusion_mat = confusion_matrix(y_test, y_pred)
print('Accuracy SVM:', accuracy)
print('Confusion Matrix:', confusion_mat)
from sklearn.metrics import accuracy_score
from sklearn import metrics
acc=(metrics.accuracy_score(y_pred,y_test)*100)
```

```

print("Accuracy is:",acc)
cm1 = metrics.confusion_matrix(y_pred,y_test)
total1=sum(sum(cm1))
import matplotlib.pyplot as plt
sensitivity1 = cm1[0,0]/(cm1[0,0]+cm1[0,1])
print('Sensitivity : ', sensitivity1 )
specificity1 = cm1[1,1]/(cm1[1,0]+cm1[1,1])
print('Specificity : ', specificity1)
import seaborn as sns
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
print("\nClassification Report\n")
print(classification_report(y_pred,y_test, target_names=['Class 1', 'Class 2']))
confusion_mtx = confusion_matrix(y_pred,y_test)
# plot the confusion matrix
f,ax = plt.subplots(figsize=(8, 8))
sns.heatmap(confusion_mtx, annot=True,
linewidths=0.01,cmap="Blues",linecolor="gray", fmt= '.1f',ax=ax)
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.title("Confusion Matrix")
plt.show()

```

### **KNN training and testinng code :**

```

import numpy as np
from sklearn.metrics import accuracy_score, confusion_matrix
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.neighbors import KNeighborsClassifier

```

```

from sklearn.model_selection import train_test_split
data = pd.read_csv('dataset.csv')
X = data['message']
y = data['label']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
vectorizer = CountVectorizer()
X_train_count = vectorizer.fit_transform(X_train)
X_test_count = vectorizer.transform(X_test)
knn = KNeighborsClassifier(n_neighbors=1)
knn.fit(X_train_count, y_train)
y_pred = knn.predict(X_test_count)
accuracy = accuracy_score(y_test, y_pred)
confusion_mat = confusion_matrix(y_test, y_pred)
print('Accuracy KNN:', accuracy)
print('Confusion Matrix:', confusion_mat)
from sklearn.metrics import accuracy_score
from sklearn import metrics
acc=(metrics.accuracy_score(y_pred,y_test)*100)
print("Accuracy is:",acc)
cm1 = metrics.confusion_matrix(y_pred,y_test)
total1=sum(sum(cm1))
import matplotlib.pyplot as plt
sensitivity1 = cm1[0,0]/(cm1[0,0]+cm1[0,1])
print('Sensitivity : ', sensitivity1 )
specificity1 = cm1[1,1]/(cm1[1,0]+cm1[1,1])
print('Specificity : ', specificity1)
import seaborn as sns
from sklearn.metrics import confusion_matrix

```

```

from sklearn.metrics import classification_report
print("\nClassification Report\n')
print(classification_report(y_pred,y_test, target_names=['Class 1', 'Class 2']))
confusion_mtx = confusion_matrix(y_pred,y_test)
# plot the confusion matrix
f,ax = plt.subplots(figsize=(8, 8))
sns.heatmap(confusion_mtx, annot=True,
linewidths=0.01,cmap="Blues",linecolor="gray", fmt= '.1f',ax=ax)
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.title("Confusion Matrix")
plt.show()

```

## Final Prediction

```

new_text = ['m planning to spend as little time as possible on the']
new_text_vectorized = vectorizer.transform(new_text)
prediction = svm.predict(new_text_vectorized)
print('Prediction:', prediction)
if int(prediction)==0:
    print("SVM RESULT -Normal")
else:
    print("SVM RESULT -Depression")
prediction1 = knn.predict(new_text_vectorized)
print('Prediction:', prediction1)
if int(prediction1)==0:
    print("KNN RESULT -Normal")
else:
    print("KNN RESULT -Depression")

```



## REFERENCES

- [1]. M. Rizwan, M. F. Mushtaq, U. Akram, A. Mehmood, I. Ashraf and B. Sahelices, "Depression Classification From Tweets Using Small Deep Transfer Learning Language Models," in *IEEE Access*, vol. 10, pp. 129176-129189, 2022, doi: 10.1109/ACCESS.2022.3223049.
- [2]. J. Zhou, H. Zogan, S. Yang, S. Jameel, G. Xu and F. Chen, "Detecting Community Depression Dynamics Due to COVID-19 Pandemic in Australia," in *IEEE Transactions on Computational Social Systems*, vol. 8, no. 4, pp. 982-991, Aug. 2021, doi: 10.1109/TCSS.2020.3047604.
- [3]. R. S. Jagdale and S. S. Deshmukh, "Sentiment Classification on Twitter and Zomato Dataset Using Supervised Learning Algorithms," 2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing (ICSIDEMPC), Aurangabad, India, 2020, pp. 330-334, doi: 10.1109/ICSIDEMPC49020.2020.9299582.
- [4]. R. S. Skaik and D. Inkpen, "Predicting Depression in Canada by Automatic Filling of Beck's Depression Inventory Questionnaire," in *IEEE Access*, vol. 10, pp. 102033-102047, 2022, doi: 10.1109/ACCESS.2022.3208470.
- [5]. Safa, R., Bayat, P. & Moghtader, L. Automatic detection of depression symptoms in twitter using multimodal analysis. *J Supercomput* 78, 4709–4744 (2022).
- [6]. Sancheng Peng, Lihong Cao, Yongmei Zhou, Zhouhao Ouyang, Aimin Yang, Xinguang Li, Weijia Jia, Shui, A survey on deep learning for textual emotion analysis in social networks, *Digital Communications and Networks*, Volume 8, Issue 5, 2022, Pages 745-762, ISSN 2352-8648,
- [7]. M. M. Tadesse, H. Lin, B. Xu and L. Yang, "Detection of Depression-Related Posts in Reddit Social Media Forum," in *IEEE Access*, vol. 7, pp. 44883-44893, 2019, doi: 10.1109/ACCESS.2019.2909180.

- [8]. Mohammed Hasan Ali Al-Abyadh, Mohamed A. M. Iesa, Hani Abdel Hafeez Abdel Azeem, Devesh Pratap Singh, Pardeep Kumar, Mohamed Abdulamir, Asadullah Jalali, "Deep Sentiment Analysis of Twitter Data Using a Hybrid Ghost Convolution Neural Network Model", *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 6595799, 8 pages, 2022.
- [9]. H. T. Phan, V. C. Tran, N. T. Nguyen and D. Hwang, "Improving the Performance of Sentiment Analysis of Tweets Containing Fuzzy Sentiment Using the Feature Ensemble Model," in *IEEE Access*, vol. 8, pp. 14630-14641, 2020, doi: 10.1109/ACCESS.2019.2963702.
- [10]. Jiang C, Li Y, Tang Y, Guan C. Enhancing EEG-Based Classification of Depression Patients Using Spatial Information. *IEEE Trans Neural Syst Rehabil Eng*. 2021;29:566-575. doi: 10.1109/TNSRE.2021.3059429. Epub 2021 Mar 3. PMID: 33587703.
- [11]. Sonam Gupta, Lipika Goel, Arjun Singh, Ajay Prasad, Mohammad Aman Ullah, "Psychological Analysis for Depression Detection from Social Networking Sites", *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 4395358, 14 pages, 2022.
- [12]. Khatua, Aparup & Cambria, Erik & Ghosh, Kuntal & Chaki, Nabendu & Khatua, Apalak. (2019). Tweeting in Support of LGBT?: A Deep Learning Approach. 342-345. 10.1145/3297001.3297057.
- [13]. Angskun, J., Tipprasert, S. & Angskun, T. Big data analytics on social networks for real-time depression detection. *J Big Data* 9, 69 (2022).
- [14]. Kayıkçı, Ş. SenDemonNet: sentiment analysis for demonetization tweets using heuristic deep neural network. *Multimed Tools Appl* 81, 11341–11378 (2022).
- [15]. Anu Priya, Shruti Garg, Neha Prerna Tigga, Predicting Anxiety, Depression and Stress in Modern Life using Machine Learning Algorithms, *Procedia Computer Science*, Volume 167, 2020, Pages 1258-1267, ISSN 1877-0509

- [16]. Jiang C, Li Y, Tang Y, Guan C. Enhancing EEG-Based Classification of Depression Patients Using Spatial Information. *IEEE Trans Neural Syst Rehabil Eng.* 2021;29:566-575. doi: 10.1109/TNSRE.2021.3059429. Epub 2021 Mar 3. PMID: 33587703.
- [17] X. Li, H. Zhang, and X.-H. Zhou, “Chinese clinical named entity recognition with variant neural structures based on BERT methods,” *J. Biomed.Informat.*, vol. 107, Jul. 2020, Art. no. 103422.
- [18] Y. Chang, L. Kong, K. Jia, and Q. Meng, “Chinese named entity recognition method based on BERT,” in *Proc. IEEE Int. Conf. Data Sci. Comput. Appl. (ICDSCA)*, Oct. 2021, pp. 294–299.
- [19] N. Jalal, A. Mehmood, G. S. Choi, and I. Ashraf, “A novel improved random forest for text classification using feature ranking and optimal number of trees,” *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 6, pp. 2733–2742, Jun. 2022.
- [20] V. Rupapara, F. Rustam, A. Amaar, P. B. Washington, E. Lee, and I. Ashraf, “Deepfake tweets classification using stacked bi-LSTM and words embedding,” *PeerJ Comput. Sci.*, vol. 7, Oct. 2021, Art. no. e745.
- [21] C. Wu, F. Wu, and Y. Huang, “One teacher is enough? Pre-trained language model distillation from multiple teachers,” 2021, arXiv:2106.01023.
- [22] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter,” 2019, arXiv:1910.01108.
- [23] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, “TinyBERT: Distilling BERT for natural language understanding,” 2019, arXiv:1909.10351.
- [24] W. Zhang, L. Hou, Y. Yin, L. Shang, X. Chen, X. Jiang, and Q. Liu, “TernaryBERT: Distillation-aware ultra-low bit BERT,” 2020, arXiv:2009.12812.

- [25] Y. Xu, X. Qiu, L. Zhou, and X. Huang, “Improving BERT fine-tuning via self-ensemble and self-distillation,” 2020, arXiv:2002.10345.
- [26] S. Sun, Y. Cheng, Z. Gan, and J. Liu, “Patient knowledge distillation for BERT model compression,” 2019, arXiv:1908.09355.
- [27] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, “LightMBERT: A simple yet effective method for multilingual BERT distillation,” 2021, arXiv:2103.06418.
- [28] S. Madichetty, S. Muthukumarasamy, and P. Jayadev, “Multi-modal classification of Twitter data during disasters for humanitarian response,” *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 11, pp. 10223–10237, Nov. 2021.