

USING MACHINE LEARNING TO DETECT DEPRESSION

A Project Report submitted in partial fulfillment of the requirements for the award of

degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

Mantravadi Nikhita (221910302031)

Vora Sreeja(221910302056)

B. Sridhar (221910302006)

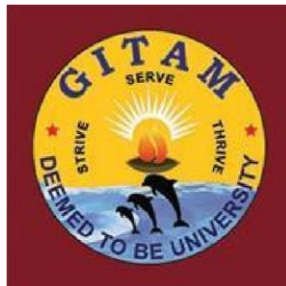
B. Ritwin (221910302007)

P. V. Jayanth Phani (221910302044)

Under the esteemed guidance of

Mrs. G. Karthika

Asst.Professor



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GITAM

(Deemed to be University)

HYDERABAD

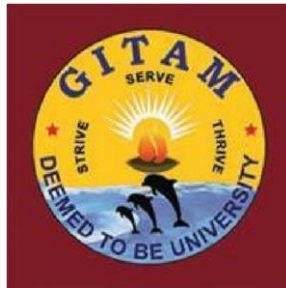
APRIL, 2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GITAM SCHOOL OF TECHNOLOGY

GITAM

(Deemed to be University)



DECLARATION

We, hereby declare that the project report entitled “**USING MACHINE LEARNING TO DETECT DEPRESSION**” is an original work done in the Department of Computer Science and Engineering, GITAM School of Technology, GITAM (Deemed to be University) submitted in partial fulfillment of the requirements for the award of the degree of B.Tech. in Computer Science and Engineering. The work has not been submitted to any other college or University for the award of any degree or diploma.

Date:

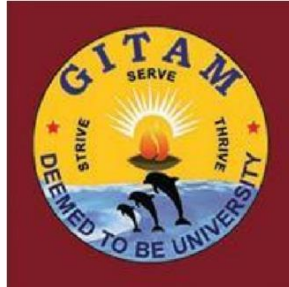
Registration No(s).	Name(s)	Signature(s)
221910302031	Mantravadi Nikhita	
221910302056	Vora Sreeja	
221910302006	B. Sridhar	
221910302007	B. Ritwin	
221910302044	P.V. Jayanth Phani	

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GITAM SCHOOL OF TECHNOLOGY

GITAM

(Deemed to be University)



CERTIFICATE

This is to certify that the project report entitled “**USING MACHINE LEARNING TO DETECT DEPRESSION**” is a bonafide record of work carried out by **MANTRAVADI NIKHITA(221910302031), VORA SREEJA(221910302056), B. SRIDHAR(221910302006), B. RITWIN(221910302007), P.V. JAYANTH PHANI(221910302044)** students submitted in partial fulfillment of requirement for the award of degree of Bachelors of Technology in Computer Science and Engineering.

Project Guide

Mrs. G. Karthika
Asst. Professor
CSE Dept.

Project Coordinator

Dr. S. Aparna
Asst. Professor
CSE Dept.

Head of the Department

Dr. K. Sudeep
Professor
CSE Dept.

TABLE OF CONTENTS:

ABSTRACT	i
LIST OF FIGURES	ii
CHAPTER-1: INTRODUCTION	1
1.1 MOTIVATION	2
1.2 PROBLEM IDENTIFICATION	3
1.3 OBJECTIVE	4
1.4 LIMITATIONS	5
1.5 OUTCOMES	6
1.6 APPLICATIONS	7
CHAPTER-2: LITERATURE SURVEY	8
CHAPTER-3: PROBLEM ANALYSIS	16
3.1 PROBLEM STATEMENT	16
3.2 EXISTING SYSTEM	16
3.3 FLOWS AND DISADVANTAGES	17
3.4 PROPOSED SYSTEM	18
3.5 FUNCTIONAL REQUIREMENTS	19
3.6 NON- FUNCTIONAL REQUIREMENTS	20
CHAPTER-4: SYSTEM DESIGN	21
4.1 PROPOSED SYSTEM ARCHITECTURE	21
4.1.1 ARCHITECTURE DESIGN FOR RANDOM FOREST	21
4.1.2 ARCHITECTURE DESIGN FOR GAUSSIAN NAÏVE BAYES	22
4.1.3 ARCHITECTURE DESIGN FOR DECISION TREE	23
4.1.4 ARCHITECTURE DESIGN FOR K-NEAREST NEIGHBOUR	24
4.1.5 ARCHITECTURE DESIGN FOR MLP CLASSIFIER	25
4.2 UML DIAGRAMS	26
4.2.1 ADVANTAGES	26
4.2.2 USE CASE DIAGRAM	27
4.2.3 CLASS DIAGRAM	28
4.2.4 ACTIVITY DIAGRAM	29
4.2.5 SEQUENCE DIAGRAM	30
CHAPTER-5: IMPLEMENTATION	31
5.1 OVERVIEW OF TECHNOLOGIES	31
5.1.1 PYTHON	31
5.1.2 GOOGLE COLAB	31
5.1.3 GOOGLE DRIVE	32
5.2 WORKFLOW	32
5.2.1 IMPORTING LIBRARIES	32

5.2.3 EXPLORATORY DATA ANALYSIS	34
5.2.4 DATA PROCESSING	38
5.2.5 SPLITTING DATASET INTO TRAINING AND TEST SET	42
5.2.6 APPLYING MACHINE LEARNING MODELS	43
5.3 LIBRARIES IMPORTED	47
5.3.1 PANDAS	47
5.3.2 SEABORN	47
5.3.3 NUMPY	48
5.3.4 SCIKIT-LEARN	48
5.4 DATASET	48
5.4.1 DATA COLLECTION	48
5.4.2 FINAL DATA	50
5.5 ALGORITHMS IMPLEMENTED	51
5.5.1 RANDOM FOREST	51
5.5.2 GAUSSIAN NAÏVE BAYES	51
5.5.3 DECISION TREE	51
5.5.4 K- NEAREST NEIGHBOURS	51
5.5.5 MLP CLASSIFIER	52
CHAPTER – 6: TESTING AND VALIDATION	53
6.1 SYSTEM TESTING	52
6.1.1 ACCURACY TESTING FOR RANDOM FOREST	52
6.1.2 ACCURACY TESTING FOR GAUSSIAN NAÏVE BAYES	53
6.1.3 ACCURACY TESTING FOR DECISION TREE	53
6.1.4 ACCURACY TESTING FOR KNN	54
6.1.5 ACCURACY TESTING FOR MLP CLASSIFIER	54
6.2 PERFORMANCE METRICS	55
6.2.1 PERFORMANCE METRICS FOR RANDOM FOREST	55
6.2.2 PERFORMANCE METRICS FOR GAUSSIAN NAÏVE BAYES	56
6.2.3 PERFORMANCE METRICS FOR DECISION TREE	56
6.2.4 PERFORMANCE METRICS FOR KNN	56
6.2.5 PERFORMANCE METRICS FOR MLP CLASSIFIER	57
6.3 CONFUSION MATRIX	57
6.3.1 CONFUSION MATRIX FOR RANDOM FOREST	57
6.3.2 CONFUSION MATRIX FOR GAUSSIAN NAÏVE BAYES	57
6.3.3 CONFUSION MATRIX FOR DECISION TREE	57

6.3.4 CONFUSION MATRIX FOR KNN	
6.3.5 CONFUSION MATRIX FOR MLP CLASSIFIER	58
7. CONCLUSION	59
8. REFRENECES	60

ABSTRACT:

Identification of factors that are responsible for causing depression may lead to new experiments and treatments. Because depression as a disease is becoming a leading community health concern worldwide. Depression is a common mental disorder. Globally, it is estimated that 5% of adults suffer from the disorder. It is characterized by persistent sadness and a lack of interest or pleasure in previously rewarding or enjoyable activities. It can also disturb sleep and appetite. The quality of life that a person has also deeply affects the person's mental state. Using a survey study, based on various characteristics or factors one may be able to predict if the person is Depressed or not. People in rural areas tend to be more depressed due to their living conditions. Identifying depression can be a challenge in rural areas especially as people do not tend to have any mental health awareness. The classification of depression may help people achieve the help they need. Since, depression is a very complex mental health disease to analyze, finding even the smallest patterns helps a lot in the study. By using Machine Learning algorithms to analyze and classify if the person is depressed or not can be a huge progress technologically.

GROUP MEMBERS:

M. NIKHITA- 221910302031

V. SREEJA- 221910302056

JAYANTH PHANI- 221910302044

B. RITWIN- 221910302007

B. SRIDHAR- 221910302006

LIST OF FIGURES:

1. ARCHITECTURE DIAGRAM FOR RANDOM FOREST	21
2. ARCHITECTURE DIAGRAM FOR GUASSIAN NAÏVE BAYES	22
3. ARCHITECTURE DIAGRAM FOR DECISION TREE	23
4. ARCHITECTURE DIAGRAM FOR KNN	24
5. ARCHITECTURE DIAGRAM FOR MLP CLASSIFIER	25
6. USE CASE DIAGRAM	27
7. CLASS DIAGRAM	28
8. ACTIVITY DIAGRAM	29
9. SEQUENCE DIAGRAM	30

1. INTRODUCTION

Healthcare is one of the major problems faced by the entire world regardless of the situation whether the country is developing or developed. As a leading interest worldwide, smart, efficient, and secure healthcare systems are developed to Improve the quality of life. The early studies of human behavior have attracted the researchers of different fields to work in the discipline of psychology and Neuroscience The authors discuss the potential of these technologies to revolutionize the way we approach health research, diagnosis, and treatment. They also highlight the challenges that need to be overcome in order to fully harness the power of big data and machine learning in the health sciences. The authors emphasize the need for collaboration and integration between different fields, including computer science, statistics, biology, and medicine, to fully leverage the potential of big data and machine learning. They also discuss ethical concerns related to privacy, security, and bias, which need to be carefully addressed. Overall, the article provides an insightful overview of the current state and future prospects of big data and machine learning in the health sciences, and highlights the importance of interdisciplinary collaboration and responsible use of these technologies. The review covered a range of mental health conditions and identified several key areas where machine learning can be applied. The study found that machine learning has the potential to improve diagnosis and treatment of mental health conditions, identify risk factors for suicide and self-harm, and enhance mental health monitoring and support. The review also highlighted the need for further research and collaboration between mental health professionals and data scientists to fully realize the potential of machine learning in mental health.

Overall, the review concluded that machine learning has the potential to significantly improve mental health outcomes and is a promising area for future research and development. However, the authors caution that while machine learning can provide valuable insights, it should not be seen as a replacement for human expertise in mental health care. Instead, the authors recommend that machine learning should be used in conjunction with traditional methods to improve mental health outcomes and provide more personalized and effective treatment. The same is the case with the growing field of research in computer science and machine learning. Identifying the mental health issues of a patient is an enduring challenge to doctors and healthcare organizations and especially among younger people, is not a new phenomenon. In their article "Persuasive technology for mental health: One step closer to (Mental health care) equality?", Kolenik and Gams examine the potential of persuasive technology to improve mental health care and promote equality in access to care. They define persuasive technology as technology designed to influence human behavior, attitudes, or emotions. The authors discuss various forms of persuasive technology, such as virtual reality, serious games, and mobile applications, and how they can be used to promote mental health and wellbeing. Gams also address potential challenges and ethical considerations

associated with the use of persuasive technology in mental health care. They emphasize the need for careful evaluation and regulation of these technologies to ensure their safety, effectiveness, and accessibility to diverse populations. Overall, the article provides a comprehensive overview of the potential of persuasive technology to improve mental health care, and highlights the importance of responsible and inclusive design and implementation. The authors argue that the increasing availability of large amounts of health data, combined with advances in machine learning techniques, offers new opportunities for understanding and improving human health. The authors begin by discussing the challenges and opportunities associated with big data in the health sciences. They note that while the sheer volume of health data can be overwhelming, advances in data analytics and machine learning can help researchers identify patterns and relationships that were previously impossible to detect. For example, machine learning algorithms can be used to analyze electronic health records to identify risk factors for disease or to predict patient outcomes. The authors also discuss some of the ethical and practical considerations associated with the use of big data and machine learning in the health sciences. They note that while these techniques offer great potential, they also raise important questions about data privacy, data ownership, and the potential for algorithmic bias. The authors conclude by calling for a collaborative approach to the use of big data and machine learning in the health sciences, one that involves researchers, clinicians, policymakers, and patients.

1.1 MOTIVATION:

Depression is a widespread mental health disorder that affects millions of people worldwide. It can negatively impact various aspects of an individual's life, including their quality of life and motivation. Machine learning (ML) can be used to investigate the association between depression and quality of life motivation in healthcare systems. This approach can help healthcare providers identify patients who are at high risk of depression and provide early intervention to improve their quality of life and motivation.

One way to investigate the association between depression and quality of life motivation using ML is by predictive models. Predictive models can be trained using patient data, including demographic information, medical history, and mental health assessments. The trained model can then be used to predict the probability of depression and low quality of life motivation for new patients.

Another approach to using ML in healthcare systems to investigate the association between depression and quality of life motivation is through the use of natural language processing (NLP). NLP can be used to analyze patient data, including medical records, social media posts, and online forums, to identify patterns and correlations between depression and quality of life motivation.

ML can also be used in healthcare systems to develop personalized treatment plans for patients with depression and low quality of life motivation. ML algorithms can analyze patient data, including medication history and mental health assessments, to identify the most effective treatment plan for each patient. This approach can help healthcare providers tailor treatment plans to the individual needs of each patient and improve their overall quality of life and motivation.

In conclusion, ML can be a powerful tool in healthcare systems for investigating the association between depression and quality of life motivation. By using predictive models, NLP, and personalized treatment plans, healthcare providers can identify patients at high risk of depression, analyze patient data to identify patterns and correlations, and develop personalized treatment plans to improve patient outcomes.

1.2 PROBLEM IDENTIFICATION:

Depression is a major public health problem that affects millions of people worldwide. It is a complex mental health disorder that can significantly impact an individual's quality of life and motivation. Depression can lead to a range of physical, emotional, and cognitive symptoms, including persistent sadness, low mood, lack of energy, sleep disturbances, and loss of interest in daily activities. These symptoms can affect an individual's ability to function at work, school, and in their personal life.

One of the main challenges in treating depression is identifying patients who are at high risk of developing the disorder. Many patients with depression may not seek help due to stigma or lack of awareness of the symptoms. Moreover, depression can often be misdiagnosed or overlooked in primary care settings, leading to delayed treatment and poor outcomes.

Another challenge in treating depression is the lack of effective treatment options. Traditional treatments for depression, such as medication and therapy, may not be effective for all patients, and some patients may experience significant side effects. Additionally, there is a need for personalized treatment options that are tailored to the individual needs of each patient.

Machine learning (ML) has the potential to address some of these challenges by providing a data-driven approach to identifying patients at risk of depression and developing personalized treatment plans. ML algorithms can analyze large amounts of patient data, including medical records, social media posts, and online forums, to identify patterns and correlations between depression and quality of life. This approach can help healthcare providers identify patients who are at high risk of developing depression and provide early intervention to prevent the disorder from worsening.

However, there are also challenges associated with implementing ML-based healthcare systems. One challenge is the need for high-quality data that is representative of the patient population. Additionally, there is a need for robust privacy and security measures to protect patient data. There is also a need for healthcare providers to have the necessary skills and training to use ML algorithms effectively.

1.3 OBJECTIVES:

- Develop a database of patients with depression and quality of life measures, including demographic and clinical characteristics.
- Use machine learning algorithms to analyze the database and identify patterns and associations between depression and quality of life measures.
- Develop predictive models to identify patients who are at risk for poor quality of life outcomes based on their depression symptoms.
- Develop algorithms to identify effective treatments for patients with depression that lead to improved quality of life outcomes.
- Use natural language processing techniques to extract useful information from clinical notes and other unstructured data sources.
- Use deep learning models to analyze brain imaging data to identify neurobiological markers of depression and their relationship with quality of life.
- Evaluate the efficacy of various treatment options for depression and their impact on quality of life using machine learning-based clinical trials.
- Develop personalized treatment plans for patients with depression based on their individual characteristics and predicted outcomes.
- Develop a user-friendly interface for healthcare providers to access and interpret the results of the machine learning algorithms.
- Continuously update and refine the machine learning models and database based on new data and research findings to improve the accuracy and utility of the system.

1.4 LIMITATIONS:

- Lack of data: The system may not have access to a sufficient amount of high-quality data to accurately identify patterns and associations between depression and quality of life measures.
- Bias in data: The data used to train the machine learning models may contain bias due to factors such as sampling or selection bias, leading to inaccurate results.

- **Generalizability:** The results of the machine learning models may not be generalizable to other populations or settings due to differences in patient characteristics, treatment options, and other factors.
- **Data privacy and security:** There may be concerns about the privacy and security of patient data used to train and run the machine learning models.
- **Interpretability:** The machine learning models may be difficult to interpret, making it challenging for healthcare providers to understand and act on the results.
- **Cost:** Developing and implementing a machine learning-based healthcare system can be costly, which may limit its accessibility and adoption.
- **Ethical considerations:** There may be ethical considerations around the use of machine learning in healthcare, such as issues around informed consent and potential harm to patients.
- **Technical limitations:** The system may be limited by technical factors such as computational power and infrastructure, leading to slow performance or inaccurate results.
- **Incomplete data:** Patient data may be incomplete or inconsistent, leading to gaps or errors in the analysis.
- **Limitations of depression assessment tools:** The accuracy of the machine learning models may be limited by the quality of the depression assessment tools used to collect patient data.

1.5 OUTCOMES

- **Define the problem:** The first step in using machine learning to detect depression-related outcomes is to define the problem. This involves identifying what specific outcomes related to depression you want to detect, such as symptoms, risk factors, or treatment effectiveness.
- **Gather and prepare data:** The next step is to gather and prepare data that can be used to train a machine learning algorithm. This may involve collecting data from surveys, medical records, or other sources, and then cleaning and organizing it in a way that is suitable for analysis.
- **Choose a machine learning algorithm:** There are many different types of machine learning algorithms, each with their own strengths and weaknesses. Some algorithms may be better suited for detecting specific types of depression-related outcomes than others, so it's important to choose an algorithm that is appropriate for your specific problem.
- **Train the algorithm:** Once you have chosen an algorithm, you will need to train it using your prepared data. This involves providing the algorithm with input data and known outcomes, and then adjusting its parameters to minimize errors in its predictions.
- **Validate the algorithm:** After training the algorithm, it is important to validate it using a separate set of data that was not used for training. This helps to ensure that the algorithm is not overfitting to the training data and can generalize to new data.

- **Feature selection:** In order to improve the accuracy of the algorithm, it may be necessary to select a subset of features that are most relevant to the problem at hand. This can help to reduce noise and improve the quality of the algorithm's predictions.
- **Model evaluation:** Once you have trained and validated your algorithm, you will need to evaluate its performance using metrics such as accuracy, precision, and recall. This can help you to determine whether the algorithm is accurate enough for practical use.
- **Hyperparameter tuning:** In order to optimize the performance of the algorithm, it may be necessary to adjust its hyperparameters. This involves adjusting parameters such as learning rate, regularization strength, or number of hidden layers to improve the algorithm's accuracy.
- **Deployment:** Once you are satisfied with the performance of the algorithm, you can deploy it in a production environment. This may involve integrating it into a larger system or creating a standalone application that can be used to detect depression-related outcomes.
- **Ongoing monitoring and improvement:** Even after deployment, it is important to monitor the performance of the algorithm and make improvements as necessary. This may involve retraining the algorithm with new data or adjusting its parameters to improve its accuracy over time.

1.6 APPLICATIONS:

Depression is a mental health condition that affects millions of people worldwide. It is a complex disorder that is often difficult to diagnose and treat. Machine learning methods have shown great promise in detecting depression, enabling early intervention and treatment. In this article, we will discuss the use of machine learning methods in detecting depression. Machine learning algorithms are statistical models that learn patterns and relationships from data. They can be used to analyze large amounts of data and identify patterns that are not visible to the human eye. This makes them well-suited for the detection of depression, which can be characterized by subtle changes in behavior and speech patterns.

One of the most promising machine learning techniques for detecting depression is natural language processing (NLP). NLP algorithms can analyze text data, such as social media posts or electronic medical records, to identify patterns in language use that are associated with depression. For example, individuals with depression may use more negative words or express fewer positive emotions in their writing.

Another promising approach is the use of machine learning algorithms to analyze facial expressions and body language. Researchers have found that people with depression display distinct facial expressions that can be detected by machine learning algorithms. This technology could be used to develop a depression screening tool that analyzes a person's facial expressions in real-time.

Machine learning can also be used to analyze physiological data, such as heart rate variability, sleep patterns, and activity levels, to detect depression. This approach has shown promise in several studies, with machine learning algorithms achieving high accuracy rates in identifying individuals with depression.

A key advantage of machine learning algorithms is their ability to learn and adapt to new data. As more data is collected and analyzed, the algorithms can refine their models and improve their accuracy. This makes them ideal for developing personalized depression detection tools that can be tailored to an individual's specific symptoms and risk factors.

In conclusion, machine learning methods have shown great promise in detecting depression. By analyzing data from a variety of sources, including natural language, facial expressions, and physiological data, machine learning algorithms can identify patterns that are associated with depression. This technology has the potential to improve the accuracy and speed of depression detection, enabling earlier intervention and treatment. However, further research is needed to validate the efficacy of machine learning-based depression detection tools and to ensure their ethical use.

2. LITERATURE SURVEY :

W.Meng(etal)

This study is conducted on the National Health and Nutrition Examination Survey (NHANES) 2015-2016 data. The NHANES data comprise of series of health and nutrition surveys available online for researchers and data users all over the world. This survey covers physical examinations performed in mobile examination centers as well as questionnaire-based personal interviews at respondents' homes. In 2015-2016 NHANES survey data, a total of 15327 individual were selected for analysis from 30 diverse survey sites. The National Health and Nutrition Examination Survey (NHANES) is a program conducted by the Centers for DiseaseControl and Prevention (CDC) to assess the health and nutritional status of the US population. The main advantage of this survey is unique in that it combines interviews, physical examinations, and laboratory tests to obtain a comprehensive understanding of the health of the American population. NHANES provides valuable data on a wide range of health topics, including obesity, diabetes, cardiovascular disease, infectious diseases, and environmental exposures. [1]

Y. Dong(etal)

The main purpose is to model and capture the longitudinal and social aspects of a disease, the consolidation of heterogenous data is considered as essential. Data is stored in different format, with different identification; at different locations and has different quality. So, there is a need of some methods that help in consolidating the data and enable their analysis for research purposes. To normalize duplicate records from multiple sources, it is important to establish a standard data model that defines the structure of the data. This can be achieved by creating a common set of fields for each record that are consistent across all sources. Once the data model has been established, data integration tools can be used to consolidate the data from multiplesources into a single, unified database. It is also important to establish data governance policies and procedures to maintain the integrity of the data over time but its practically does not prove . Ultimately, effective data normalization from multiple sources is essential to ensure that businesses have access to reliable data that can support informed decision-making and drive successful outcomes.[2]

M. Masseroli(etal)

Unmapped , this terminology main source that there are no direct relations between any question from one data file with another data file. That why those questions were taken independently for further processing. Further to uniquely identify each relation in the given data the concept of the Secure Hash Algorithm 1 (SHA-1) function is adopted. Main advantage is integrating and querying these annotations, researchers can gain insights into complex biological systems and identify potential drug targets. Once the data has been integrated, researchers can use querying tools to extract specific information from the data. Querying tools allow researchers to search for specific genes, proteins, or other biological entities, and retrieve all relevant data associated with these entities. This enables researchers to identify patterns and relationships between different entities and gain insights into biological processes and pathways. Overall, the integration and querying of genomic and proteomic semantic annotations are essential for extracting meaningful insights from large biological datasets. By standardizing the way data is organized and categorized, researchers can more easily analyze and interpret complex biological systems, leading to new discoveries and advancements in the field of biomedicine.[3]

B. Ma, T. Jiang(etal)

The novel Data integration is about crucial process for organizations that need to consolidate data from multiple sources and make it available for analysis. However, data integration can be a challenging task because data is often stored in different formats, uses different terminologies and has different structures. The unified concept model summary is a representation of the key concepts and relationships that exist in the data being integrated. It is created by mapping the data to a common ontology or semantic model, which provides a standard vocabulary and structure for representing concepts and relationships. The second step is to extract and transform the data from each source into a format that can be integrated with the other sources. The third step is to integrate the data using the unified concept model summary, which ensures that the data is consistent and can be analyzed together. The benefits of using a data integration framework based on the unified concept model summary are numerous. It reduces the time and effort required for data integration, enables data to be integrated from multiple sources, and provides a consistent and accurate view of the data.[4]

T. Kohonen(eatl)

Self-organized formation of topologically correct feature maps is a process where neurons in the brain organize themselves in a way that reflects the structure of sensory input. This process was first described by T. Kohonen in the 1980s and has since been studied extensively in neuroscience and artificial intelligence research. The basic idea is that neurons that respond to similar features in the input will become connected, forming clusters that reflect the topological structure of the input. SOMs are composed of two layers of neurons: an input layer and a competitive layer. A study aimed to demonstrate the feasibility and effectiveness of using SOMNet to identify regional mental health needs and plan for appropriate services. The SOMNet algorithm was used to cluster the data into distinct groups based on their mental health needs. By using SOMNet to cluster data into distinct groups, mental health service providers can identify areas of need and develop targeted interventions to address those needs. This approach could help to ensure that mental health resources are allocated in an effective and efficient manner, leading to improved outcomes for service users. The self-organizing process has been shown to be a powerful mechanism for unsupervised learning in a wide range of applications, including image and speech recognition, data visualization, and clustering [5].

E. C. Dragut(eatl)

Self-organized formation of topologically correct feature maps is a process where neurons in the brain organize themselves in a way that reflects the structure of sensory input. This process was first described by T. Kohonen in the 1980s and has since been studied extensively in neuroscience and artificial intelligence research. The basic idea is that neurons that respond to similar features in the input will become connected, forming clusters that reflect the topological structure of the input. The SOMNet algorithm was used to cluster the data into distinct groups based on their mental health needs. By using SOMNet to cluster data into distinct groups, mental health service providers can identify areas of need and develop targeted interventions to address those needs. This approach could help to ensure that mental health resources are allocated in an effective and efficient manner, leading to improved outcomes for service users. The self-organizing process has been shown to be a powerful mechanism for unsupervised learning in a wide range of applications, including image and speech recognition, data visualization, and clustering [6].

S. Pighin(eatl)

Posterior probability is a fundamental concept in Bayesian statistics that provides a way to update our beliefs or knowledge about a parameter or hypothesis based on new data. In essence, posterior probability is the updated probability of a hypothesis or parameter after taking into account new data. In other words, posterior probability takes into account both our prior belief and the new evidence. One main important feature of posterior probability is that it provides a measure of uncertainty or variability in our estimate of the parameter or hypothesis. The posterior distribution describes the range of plausible values for the parameter or hypothesis based on the available data. If the prior distribution is wide or vague and the likelihood function is less informative, then the posterior distribution will be wider and less concentrated. Finally, posterior probability can be used for hypothesis testing and model comparison. Comparing the posterior probabilities of different hypotheses or models, we can determine which one is more likely given the available data. We can also use posterior probabilities to calculate the probability of specific events or outcomes, such as the probability that a treatment is effective or the probability that a stock will increase in value. Overall, posterior probability is a powerful and flexible tool that can be used to update our beliefs and make decisions. We can also use posterior probabilities to calculate the probability of specific events or outcomes, such as the probability that a treatment is effective or the probability that a stock will increase in value. Overall, posterior probability is a powerful and flexible tool that can be used to update our beliefs and make informed decisions in a wide range of applications.[7]

M. Milic(eatl)

Tobacco smoking is a major public health concern globally and has been linked to various health problems, including cancer, cardiovascular disease, and respiratory problems. Several studies have investigated the impact of smoking on health-related quality of life (HRQoL). Depression is also a major public health issue that has been linked to poor HRQoL. Studies have shown that depression is more common among smokers than non-smokers. The relationship between smoking, depression, and HRQoL among university students has been investigated in several studies. Some studies have found that depression mediates the relationship between smoking and HRQoL among university students. The mediating effect of depression means that the relationship between smoking and HRQoL is partially explained by depression. This implies that smoking may lead to depression, which in turn affects HRQoL. Alternatively, depression may lead to smoking, which then affects HRQoL [8].

C. Yan(eatl)

A study conducted in China investigated the association between sleep difficulties and symptoms of depression/anxiety in adolescents, and whether responses to academic stress mediate this relationship. The study included 1,066 adolescents aged 12-18 years old, who completed self-report questionnaires assessing their sleep quality, responses to academic stress, and symptoms of depression/anxiety. The advantages of the study showed that sleep difficulties were positively associated with symptoms of depression/anxiety in Chinese adolescents. Additionally, responses to academic stress were found to partially mediate this relationship. This suggests that the way in which adolescents respond to academic stress may be an important factor in the link between sleep difficulties and mental health outcomes. The study also found that gender differences existed in the relationship between sleep difficulties and symptoms of depression/anxiety. Overall, the findings of this study suggest that addressing responses to academic stress may be an important target for interventions aimed at reducing the negative mental health outcomes associated with sleep difficulties in Chinese adolescents. Furthermore, the results suggest that interventions may need to be tailored to account for gender differences in the relationship between sleep difficulties and mental health outcomes [9].

M. R. Miller(eatl)

The study titled "Romantic and sexual activities, parent-adolescent stress, and depressive symptoms among early adolescent girls" explored the relationship between romantic and sexual activities, parent-adolescent stress, and depressive symptoms in early adolescent girls. Girls who engage in these activities at an early age are more likely to experience stress and depression, and have a more difficult relationship with their parents. The research also revealed that parent-adolescent stress plays a crucial role in the development of depressive symptoms among young girls. When parents have a strained relationship with their adolescent daughter, the girl is more likely to engage in romantic and sexual activities as a way to cope with stress. However, this coping mechanism can ultimately lead to negative consequences for their mental health. Moreover, the study found that early adolescent girls who engage in romantic and sexual activities are more prone to experiencing depressive symptoms. Additionally, these negative emotional experiences can further damage the parent-adolescent relationship, creating a vicious cycle of stress, depression, and conflict. Overall, the study highlights the importance of early intervention and support for early adolescent girls who may be experiencing stress, depression, or engaging in risky romantic and sexual behaviours. [10]

J.-P. Lépine(eatl)

The article titled "The increasing burden of depression" explores the growing prevalence and impact of depression worldwide. Depression is a common mental disorder that can have severe consequences for individuals and society. The article highlights the increasing burden of depression, including its causes, risk factors, and consequences. The article discusses the rising prevalence of depression, which is estimated to affect more than 264 million people worldwide. Depression is a leading cause of disability and a significant contributor to the global burden of disease. The article suggests that the rising prevalence of depression may be due to a combination of genetic, environmental, and social factors. The article also highlights the risk factors associated with depression, including poverty, unemployment, social isolation, and chronic illness. These risk factors are especially prevalent in low- and middle-income countries, where depression often goes untreated due to lack of resources and stigma surrounding mental health. Overall, the article emphasizes the need for increased awareness and resources to address the growing burden of depression. The article suggests that a comprehensive approach is needed to address the complex causes and consequences of depression, including increased investment in mental health services, social support programs, and addressing the social determinants of health [11].

Authors: S. Y. Kim(eatl)

Gender and age differences can play a significant role in the experience and manifestation of depression. Research suggests that women are more likely to experience depression than men, with some studies indicating that women may be twice as likely to develop depression as men. Women may also be more likely to seek treatment for depression than men, which can contribute to the perception that depression is more prevalent among women. Age can also be a significant factor in depression. While depression can affect individuals of all ages, it is more commonly diagnosed in individuals over the age of 65. This may be due in part to the challenges and losses associated with aging, including Younger individuals may also experience depression differently than older adults. For example, teenagers and young adults may be more likely to experience irritability, anger, and substance use as symptoms of depression, while older adults may be more likely to experience physical symptoms such as fatigue and sleep disturbances. Additionally, depression in younger individuals may be more closely linked to social and environmental factors, such as bullying, academic stress, and family conflict. It is important to recognize that depression can impact individuals of all genders and ages, and that seeking treatment is crucial for recovery. Treatment options may vary depending on the individual's age and gender, as well as their unique needs and circumstances. [12]

The paper titled "Semi-supervised approach to monitoring clinical depressive symptoms in social media" presents a novel approach to detecting and monitoring clinical depressive symptoms using social media data. The approach is semi-supervised, which means that it combines labeled data (data that is manually annotated with labels indicating the presence or absence of depressive symptoms) with unlabeled data (data that is not annotated). This approach is particularly useful in the context of depression, where labeled data is often scarce. The proposed approach involves two steps: first, a classifier is trained using labeled data to identify posts that contain depressive symptoms. Second, the classifier is used to label a larger set of unlabeled data, which can then be used to monitor depressive symptoms over time. The paper demonstrates that this approach can achieve high accuracy in identifying depressive symptoms in social media data, even when only a small amount of labeled data is available. Overall, this approach has the potential to provide a low-cost, scalable way to monitor depressive symptoms in real time, which could be useful for both clinicians and individuals with depression. By monitoring social media data, it may be possible to detect depressive symptoms earlier than with traditional methods and intervene before symptoms worsen. However, as with any use of social media data, there are also important ethical considerations to take into account, such as privacy and consent[13].

Z. A. Sayyed(etal)

The paper titled "Detecting linguistic traces of depression in topic-restricted text" presents a novel approach to detecting linguistic traces of depression in topic-restricted text. The approach involves using linguistic features such as sentiment, topic, and grammar to identify patterns that are indicative of depression. This approach is particularly useful in detecting depression in text that is related to specific topics, such as health or relationships. The paper uses a dataset of forum posts from a depression forum, as well as a control dataset of forum posts from a non-depression forum. The approach involves extracting linguistic features from the forum posts and then using machine learning algorithms to classify posts as either indicative of depression or not. The results show that the approach can achieve high accuracy in detecting depression in topic-restricted text, with F1 scores of over 0.8. Overall, this approach has the potential to provide a useful tool for detecting depression in text that is related to specific topics. By identifying linguistic traces of depression, it may be possible to intervene earlier and provide targeted support to individuals who may be at risk of developing depression. However, as with any use of machine learning in mental health, there are important ethical considerations to take into account, such as privacy[14]

N Balaji(eatl)

The application of machine learning algorithms to behavioral modeling for mental health is an emerging field that has the potential to revolutionize the way mental health disorders are diagnosed, monitored, and treated. Behavioral modeling involves analyzing patterns in an individual's behavior, such as movement, speech, and social interactions, to identify potential signs of mental health disorders. Machine learning algorithms can be used to automatically analyze these patterns and identify individuals who may be at risk of developing mental health disorders. One key advantage of using machine learning algorithms for behavioral modeling is that they can analyze large amounts of data quickly and accurately. This can help to identify patterns and trends that may be difficult or impossible for human clinicians to detect. In addition, machine learning algorithms can be trained using a variety of different data sources, including wearable sensors, smartphone apps, and social media data, which can provide a more comprehensive picture of an individual's behavior. Overall, the use of machine learning algorithms for behavioral modeling in mental health has the potential to improve diagnosis, monitoring, and treatment of mental health disorders. By identifying early signs of mental health disorders, clinicians may be able to intervene earlier and provide more targeted treatment. However, it is important to consider ethical issues such as privacy and consent [15]

H.Gao(eatl)

The paper titled "Quick simulation: A review of importance sampling techniques in communications systems" provides an overview of importance sampling techniques and their application in communication systems simulation. Importance sampling is a technique used to improve the efficiency of Monte Carlo simulations by sampling from a distribution that is different from the original distribution. This can be particularly useful in communication systems simulation, where the simulation of large-scale systems can be computationally expensive. The paper reviews several importance sampling techniques, including basic importance sampling, self-normalized importance sampling, and sequential importance sampling. It discusses the advantages and disadvantages of each technique, as well as their suitability for different types of simulation problems. The paper also provides examples of how importance sampling techniques have been applied in communication systems simulation, such as in the simulation of wireless communication channels. Overall, the paper highlights the importance of importance sampling techniques in improving the efficiency of communication systems simulation. By using these techniques, it may be possible to simulate large-scale systems more quickly and accurately, which can be useful in a variety of applications, such as the design and optimization of communication systems. However, it is important to carefully select the appropriate importance sampling technique for a given simulation problem, as the effectiveness of the technique can depend on the specific characteristics of the problem being simulated[16]

Pairwise classification is a technique used in machine learning for binary classification problems, where the goal is to separate data points into two classes. In pairwise classification, the problem is decomposed into a set of binary classification problems, where each problem involves distinguishing between two classes. This can be particularly useful in problems where there are multiple classes, as it allows for the use of binary classifiers that are often simpler and more efficient than multiclass classifiers. One popular technique for pairwise classification is the support vector machine (SVM), which is a type of supervised learning algorithm that can be used for classification and regression analysis. SVMs are particularly useful for pairwise classification because they are able to identify a hyperplane that maximally separates the two classes in the feature space. The hyperplane is chosen such that the margin, or distance between the hyperplane and the nearest data points of each class, is maximized. This allows for the SVM to be highly accurate in distinguishing between the two classes. In addition, SVMs are able to handle nonlinearly separable data by mapping the data into a higher-dimensional feature space, where a linear separation may be possible. Overall, the use of pairwise classification and SVMs can be a powerful tool for binary classification problems, particularly when dealing with complex or nonlinear data. However, as with any machine learning technique, it is important to carefully consider the appropriate feature space, kernel functions, and other hyperparameters to ensure the best performance of the algorithm. In addition, it is important to be aware of potential biases in the data and algorithm, and to consider ethical issues such as privacy and consent when using these techniques[13]

3. PROBLEM ANALYSIS

3.1 Problem Statement

Depression is a major mental health disorder affecting millions of people worldwide. It is often difficult to diagnose and treat due to the subjective nature of its symptoms. Traditional diagnostic methods rely on self-reporting and clinical observations, which can be time-consuming and prone to errors. Therefore, there is a need for more efficient and accurate methods of detecting depression. Machine learning techniques can be applied to analyze large datasets of biological, behavioral, and environmental factors to develop models that can accurately predict depression. The problem statement for using machine learning to detect depression is to develop an accurate and efficient algorithm that can detect depression based on relevant features, such as speech patterns, facial expressions, sleep patterns, and other behavioral and environmental factors, with the aim of providing timely an effective diagnosis and treatment for individuals suffering from depression.

3.2 Existing System

There are already existing systems that use machine learning to detect depression. Here are a few examples:

Deprexis: Deprexis is an online therapy program that uses machine learning algorithms to personalize the therapy sessions based on the individual's needs. The program uses cognitive-behavioral therapy (CBT) techniques to help individuals manage their depression symptoms.

Suicidal Ideation Detection Challenge: This is a research competition aimed at developing machine learning algorithms that can detect suicidal ideation in individuals. The competition uses a large dataset of clinical interviews and self-reporting to train and test the algorithms.

Emotion Detection: Emotion detection is a field of study that uses machine learning algorithms to identify emotions based on facial expressions, speech patterns, and physiological signals. These algorithms can be used to detect depression based on the individual's emotional state.

Predictive models for depression: There are several research studies that have developed predictive models for depression using machine learning algorithms. These models use data from various sources, such as electronic health records, social media, and smartphone apps, to predict the risk of depression and provide timely interventions.

Overall, these existing systems demonstrate the potential of machine learning in detecting and managing depression. However, further research is needed to validate the accuracy and reliability of these systems and to ensure that they are ethically and responsibly developed and deployed.

3.3 Flaws and Disadvantages

While machine learning algorithms have shown promise in detecting depression, there are also several flaws and disadvantages to be aware of. Here are a few examples:

Limited data availability: Machine learning algorithms require large amounts of data to be trained and validated. However, there may be limited data available for certain populations, such as individuals from underrepresented communities, which can lead to biased and inaccurate models.

Lack of interpretability: Machine learning algorithms are often considered "black boxes" because they are difficult to interpret and understand. This can be problematic when it comes to diagnosing and treating depression, as healthcare providers may not fully understand how the algorithm arrived at its diagnosis or recommendations.

Ethical concerns: There are several ethical concerns surrounding the use of machine learning in healthcare. For example, there may be concerns around data privacy, informed consent, and algorithmic bias. It is important to address these concerns to ensure that these systems are developed and deployed in an ethical and responsible manner.

Limited generalizability: Machine learning models developed on one population may not generalize well to other populations. This can lead to inaccurate diagnoses and recommendations, particularly for individuals from underrepresented communities.

Human intervention still required: While machine learning algorithms can assist in the diagnosis and management of depression, they cannot replace human intervention. Healthcare providers still need to be involved in the process to ensure that patients receive appropriate care and treatment.

Overall, while machine learning algorithms have the potential to improve the detection and management of depression, it is important to be aware of their limitations and address any ethical concerns associated with their use.

3.5 Proposed System

The system we are using, is to use Machine Learning to detect depression based on a few parameters that we set ourselves. We collected our own data using a google form with the following questions:

- Gender
- Are you physically active? Do you exercise often?
- Do you smoke?
- Do you drink?
- Do you eat healthy?
- Do you have a good relationship with your parents?
- Do you often feel stressed at work/college?
- Are you in a relationship?
- Do you tend to feel alone even if you have a lot of friends?
- Do you often feel like a disappointment to people
- Do you think you are depressed

Using these questions we used five machine learning algorithms to detect if the person is depressed or not.

3.6 Functional Requirements

Functional requirements are specific features or functions that a system must have to meet the needs of its users. Here are some possible functional requirements for a machine learning system designed to detect depression:

Data collection: The system should be able to collect data from various sources, such as electronic health records, social media, and smartphone apps, to analyze relevant features associated with depression.

Preprocessing and feature extraction: The system should be able to preprocess the collected data and extract relevant features for analysis, such as speech patterns, facial expressions, and sleep patterns.

Algorithm development: The system should include algorithms for analyzing the extracted features and predicting the likelihood of depression.

Diagnosis and recommendations: The system should be able to provide a diagnosis based on the results of the analysis and provide appropriate recommendations for treatment.

User interface: The system should have a user-friendly interface that allows healthcare providers and patients to easily input data, view results, and interact with the system.

Security and privacy: The system should ensure the privacy and security of user data, including complying with relevant laws and regulations.

Continuous improvement: The system should be designed to continuously improve over time through ongoing data collection and analysis, algorithm refinement, and user feedback.

These are just a few possible functional requirements for a machine learning system designed to detect depression. The specific requirements will depend on the context, users, and goals of the system.

3.7 Non-Functional Requirements

Non-functional requirements are qualities or characteristics that a system must have, but are not related to its specific functionality. Here are some possible non-functional requirements for a machine learning system designed to detect depression:

Performance: The system should be able to process large amounts of data and provide accurate diagnoses and recommendations within a reasonable amount of time.

Reliability: The system should be reliable and available at all times to ensure that healthcare providers and patients can access it when needed.

Usability: The system should be easy to use and navigate, with clear and concise instructions and feedback.

Security: The system should ensure the confidentiality, integrity, and availability of user data, with appropriate access controls, encryption, and backup procedures in place.

Scalability: The system should be scalable and able to handle an increasing number of users and data inputs over time.

Interoperability: The system should be able to communicate with other systems and share data with other healthcare providers, as needed.

Ethical considerations: The system should be developed and deployed in an ethical and responsible manner, with appropriate safeguards in place to prevent bias, discrimination, and other ethical concerns.

Compatibility: The system should be compatible with different types of devices and platforms, to ensure that it can be accessed by a wide range of users.

These are just a few possible non-functional requirements for a machine learning system designed to detect depression. The specific requirements will depend on the context, users, and goals of the system.

4. SYSTEM DESIGN

4.1 PROPOSED SYSTEM ARCHITECTURE

4.1.1 Architecture Design for Random Forest

Random Forest is an ensemble learning method that combines multiple decision trees to make a prediction. The input data is fed into the Random Forest model, which consists of a collection of decision trees. Each decision tree is built on a randomly sampled subset of the training data and a subset of the input features. The output prediction of the Random Forest model is the average (regression) or the mode (classification) of the predictions of all the individual decision trees.

The Random Forest model is trained by constructing multiple decision trees, each using a different random subset of the training data and input features. The training process involves selecting the best split point for each node in each decision tree, using a criterion such as Gini impurity or entropy.

The output prediction of the Random Forest model can be used for a variety of tasks, such as classification or regression. The Random Forest algorithm is widely used in various fields, such as finance, marketing, and healthcare, for its high accuracy and robustness.

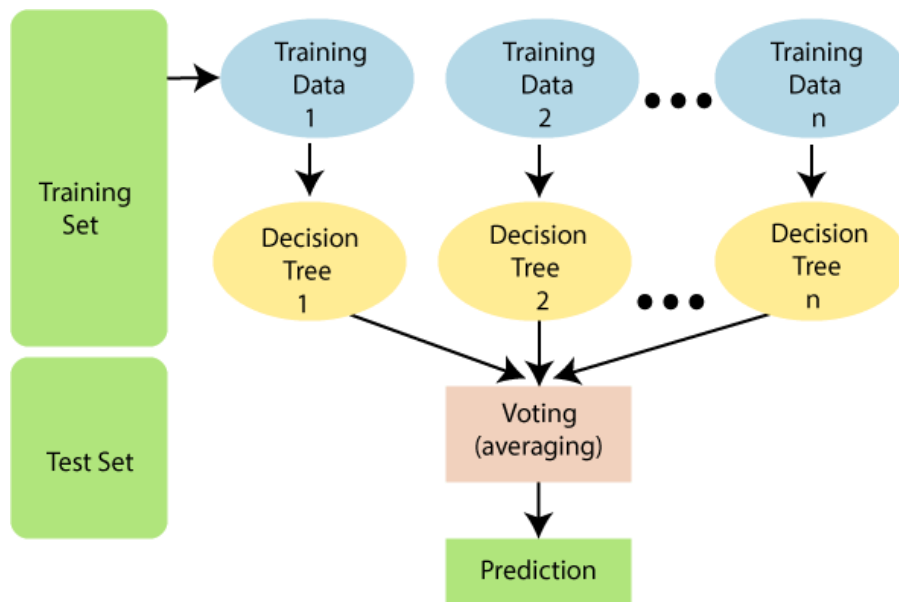


FIG 1:ARCHITECTURE DESIGN FOR RANDOM FOREST

4.1.2 Architecture Diagram for Gaussian Naïve Bayes

Gaussian Naive Bayes is a probabilistic machine learning algorithm that is commonly used for classification tasks. The input data is first preprocessed, which may include steps such as scaling, normalization, and feature extraction. The preprocessed data is then fed into the Gaussian Naive Bayes model.

The Gaussian Naive Bayes model assumes that each feature in the input data follows a Gaussian (normal) distribution. It calculates the probability of each class label given the input features using Bayes' theorem, which states that the probability of a hypothesis (in this case, the class label) given the observed evidence (the input features) is proportional to the likelihood of the evidence given the hypothesis, multiplied by the prior probability of the hypothesis.

The output prediction of the Gaussian Naive Bayes model is the class label with the highest probability. During training, the model learns the parameters of the Gaussian distributions for each feature and the prior probabilities of each class label using the training data.

The Gaussian Naive Bayes algorithm is widely used in text classification, spam filtering, and sentiment analysis, among other applications. Its simplicity and efficiency make it a popular choice for many classification tasks.

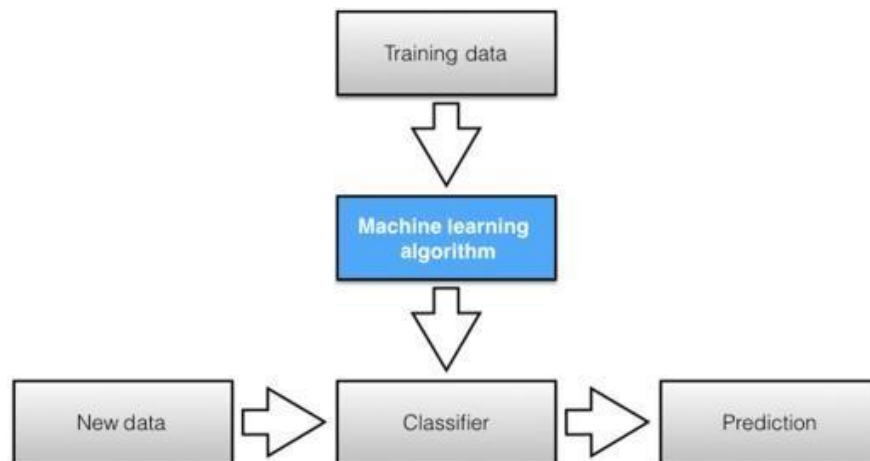


FIG-2: ARCHITECTURE DESIGN GAUSSIAN NAÏVE BAYES

4.1.3 Architecture Diagram for Decision Tree

A decision tree is a supervised learning algorithm that is used for both classification and regression tasks. The input data is fed into the decision tree model, which consists of a tree-like structure of nodes and edges. The nodes represent decision points based on the input features, and the edges represent the outcomes of the decisions.

During training, the decision tree model is built by recursively splitting the input data based on the values of the input features. At each node, the algorithm selects the best feature to split the data based on a criterion such as Gini impurity or entropy. The splitting process continues until a stopping criterion is met, such as a maximum depth or minimum number of samples in a node.

The output prediction of the decision tree model is based on the path from the root node to the leaf node that corresponds to the input data. For classification tasks, the prediction is the class label that has the highest number of samples in the corresponding leaf node. For regression tasks, the prediction is the average of the samples in the corresponding leaf node.

Decision trees are often used in combination with other algorithms, such as random forests or gradient boosting, to improve their performance. Decision trees are also interpretable, making them useful for explaining the reasoning behind a prediction.

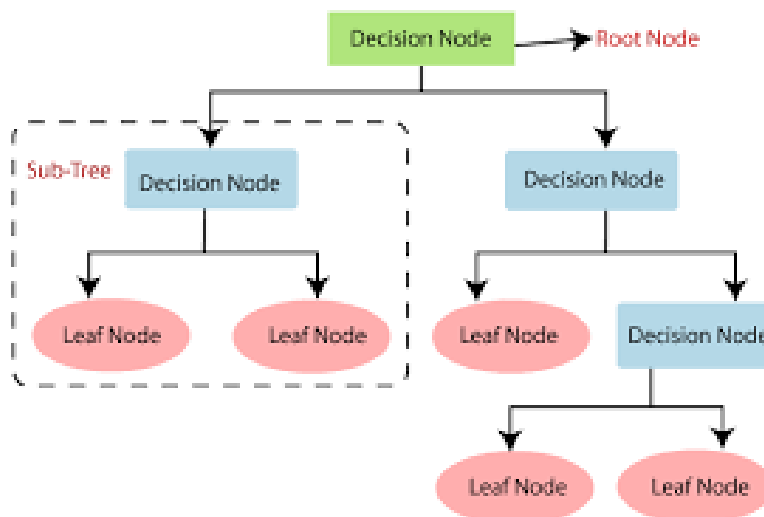


FIG 3:ARCHITECTURE DESIGN FOR DECISION TREE

4.1.4 Architecture Diagram for K-Nearest Neighbors

K-Nearest Neighbors (KNN) is a supervised learning algorithm used for both classification and regression tasks. The input data is fed into the KNN model, which stores the entire training dataset in memory.

During training, the KNN model simply memorizes the training data, so there is no specific training process as such.

When making a prediction for a new input, KNN finds the K training data points that are closest to the input data point in terms of Euclidean distance. The value of K is a hyperparameter that is set before training the model. The predicted output for the new input data is the average (regression) or mode (classification) of the output values of the K nearest neighbors.

KNN is a non-parametric algorithm, which means that it does not make any assumptions about the distribution of the data. KNN is also a lazy algorithm, meaning that it does not compute a model during training, which makes it computationally efficient.

KNN is often used in applications such as recommendation systems, image recognition, and anomaly detection. However, KNN can be computationally expensive when the training data is large, and it can be sensitive to the choice of distance metric and the value of K.

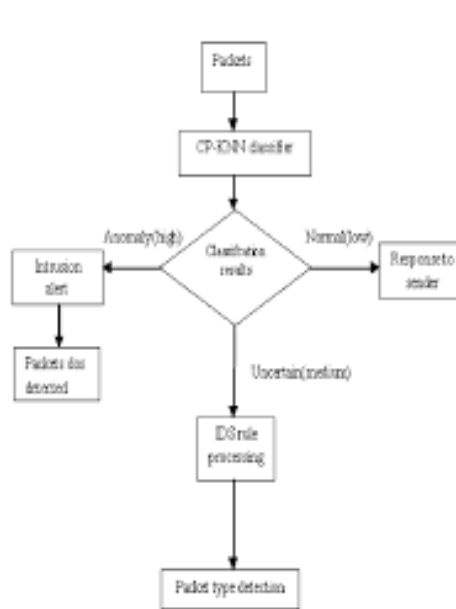


FIG 4: ARCHITECTURE DESIGN FOR KNN

4.1.5 Architecture Diagram for MLP Classifier

MLP is a feedforward neural network that is commonly used for classification and regression tasks. The input data is fed into the MLP model, which consists of multiple layers of neurons. The first layer is the input layer, which takes the input features. The last layer is the output layer, which gives the final output prediction. There may be one or more hidden layers between the input and output layers.

During training, the MLP model is trained using backpropagation, which involves adjusting the weights of the connections between the neurons to minimize a loss function. The loss function measures the difference between the predicted output and the actual output. The weights are adjusted by propagating the error backwards from the output layer to the input layer, using the chain rule of calculus to calculate the gradient of the loss function with respect to the weights.

The output prediction of the MLP model is the class label with the highest probability (for classification tasks) or the predicted value (for regression tasks). The activation function used in the output layer depends on the type of task. For example, the softmax function is commonly used for classification tasks, while the linear function is used for regression tasks.

MLP is a powerful and flexible algorithm, but it can be prone to overfitting if the model is too complex or if there is not enough training data. Regularization techniques such as L1 and L2 regularization can be used to prevent overfitting. MLP is commonly used in applications such as speech recognition, image classification, and natural language processing.

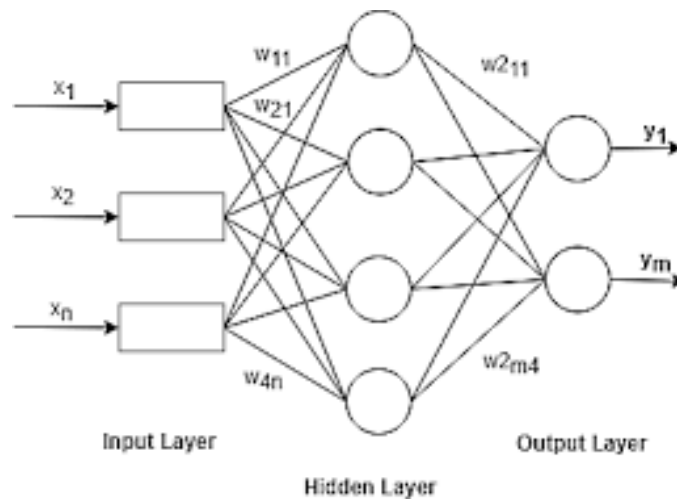


FIG-5: ARCHITECTURE DESIGN FOR MLP CLASSIFIER

4.2 UML DIAGRAMS

4.2.1 Advantages

UML diagrams have several advantages in software development, including:

Improved Communication: UML diagrams provide a visual representation of the software system, which makes it easier for developers, stakeholders, and other team members to understand and discuss the system's architecture, design, and behavior.

Effective Planning: UML diagrams can help in effective planning and decision-making. They enable developers to analyze the requirements, identify the components and their relationships, and make informed decisions about the system design.

Reusability: UML diagrams can be used to design reusable components and frameworks, which can save time and effort in future projects.

Better Maintenance: UML diagrams can help in maintaining software systems by providing a clear view of the system's structure and behavior. Developers can easily identify the areas that need modification or updates and make changes accordingly.

Simplified Documentation: UML diagrams can serve as a simplified and structured documentation of software systems, making it easier for developers to understand and maintain the system.

Effective Testing: UML diagrams can also be used for testing purposes. Developers can use the diagrams to define and plan their test cases, ensuring that the system meets the requirements and behaves as intended.

Overall, UML diagrams can be a powerful tool for developers, stakeholders, and other team members involved in software development. They can improve communication, planning, reusability, maintenance, documentation, and testing of software systems.

4.2.2 Use Case Diagram

A use case diagram is a type of UML diagram that shows the interactions between actors (users) and a system to achieve specific goals. Here is an example of a use case diagram for a machine learning system designed to detect depression in Fig 6.

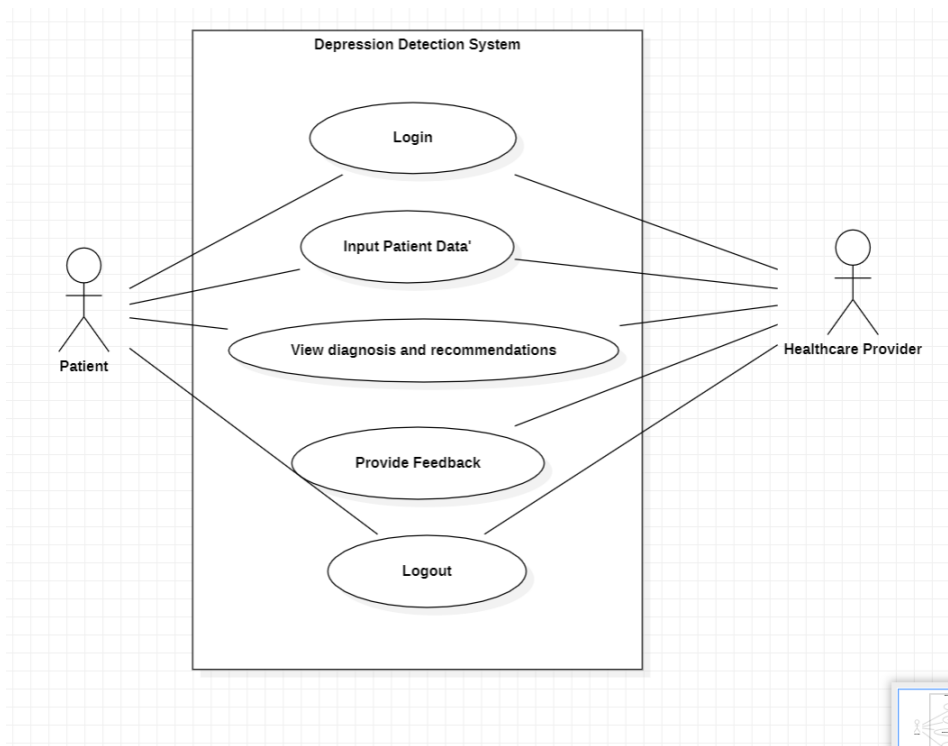


Fig - 6

In this diagram, the two actors are healthcare providers and patients. Healthcare providers can log in to the system, input patient data, view diagnoses and recommendations, provide feedback on diagnoses and recommendations, and log out. Patients can log in to the system, input personal data, view diagnoses and recommendations, provide feedback on diagnoses and recommendations, and log out.

These use cases represent the primary interactions between the actors and the system. However, there may be additional use cases or interactions depending on the specific context, users, and goals of the system.

This is just one example of a use case diagram for a machine learning system designed to detect depression. The specific actors, use cases, and interactions will depend on the context, users, and goals of the system.

4.2.3 Class Diagram

A class diagram is a type of UML diagram that shows the relationships and attributes of classes in a system. Here is an example of a class diagram for a machine learning system designed to detect depression in Fig 7

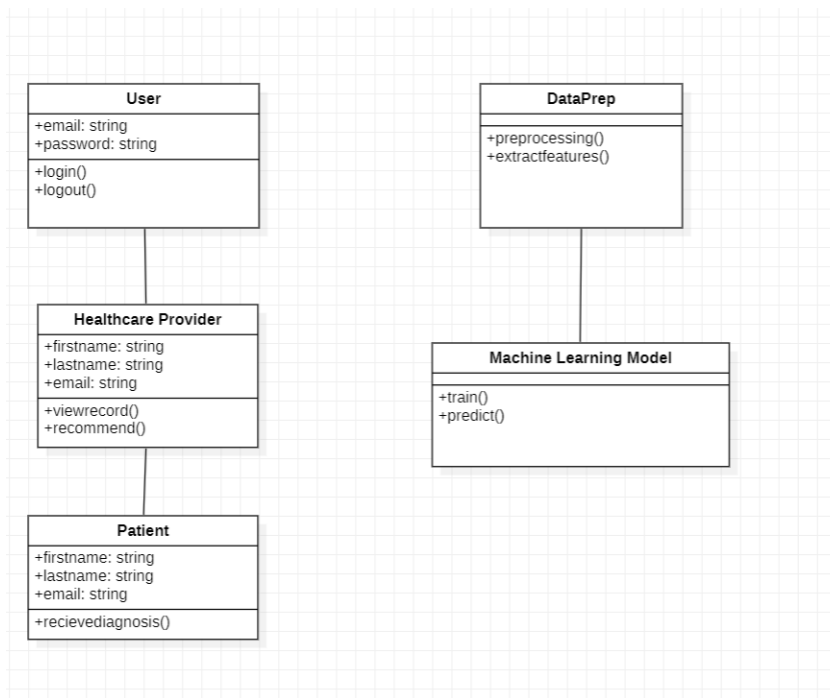


Fig-7

In this diagram, there are four classes: User, HealthcareProvider, Patient, and MachineLearningModel. The User class represents the basic attributes of a user, such as their email and password. The DataPrep class represents the data preprocessing and feature extraction functions of the system.

The HealthcareProvider class is a subclass of the User class and has additional attributes such as first name and last name. The Patient class is also a subclass of the User class and has similar attributes.

The MachineLearningModel class is responsible for training and predicting depression based on extracted features. It interacts with the DataPrep class to preprocess data and extract relevant features.

This is just one example of a class diagram for a machine learning system designed to detect depression. The specific classes, attributes, and relationships will depend on the context, users, and goals of the system.

4.2.4 Activity Diagram

An activity diagram is a type of UML diagram that shows the flow of activities and actions within a system or process. Here is an example of an activity diagram for a machine learning system designed to detect depression in Fig 8

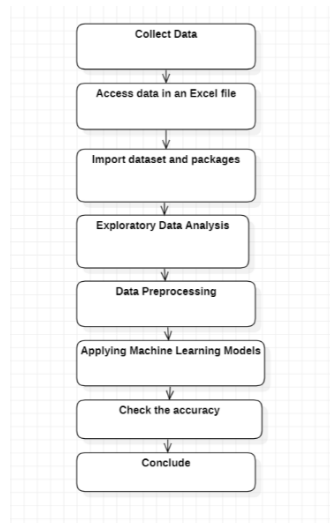


Fig-8

In this diagram, the system starts by collecting data from various sources and preprocessing it to extract relevant features. The system then develops and trains machine learning algorithm(s) to analyze the extracted features and predict the likelihood of depression. Once the system has been trained, it can input new data, preprocess it, analyze it using the trained algorithm(s), and provide a diagnosis and appropriate recommendations. The user can provide feedback on the diagnosis and recommendations, which the system can use to update the algorithm(s) and model(s) as part of a continuous improvement process. Finally, the process ends.

This is just one example of an activity diagram for a machine learning system designed to detect depression. The specific activities and flow of the system will depend on the context, users, and goals of the system.

4.2.5 Sequence Diagram

A sequence diagram is a type of UML diagram that shows the interactions between objects in a system over time. Here is an example of a sequence diagram for a machine learning system designed to detect depression in Fig 9.

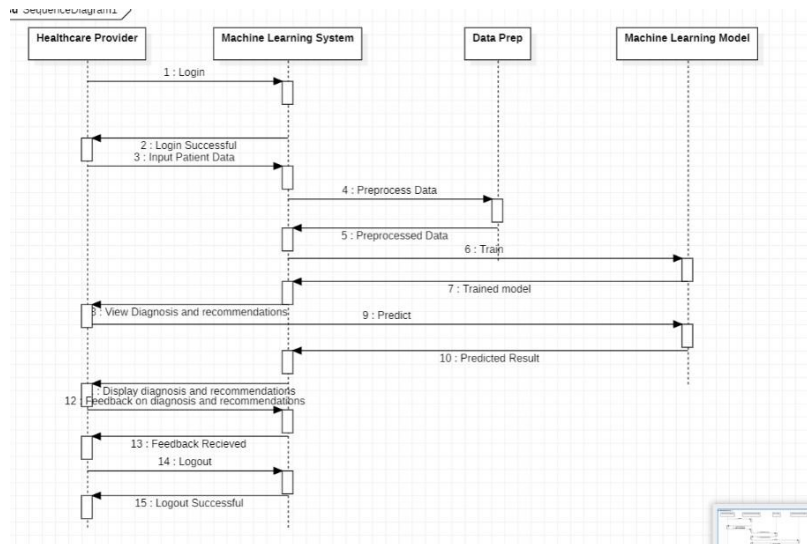


Fig 9

In this diagram, a healthcare provider logs in to the system, inputs patient data, views the diagnosis and recommendations, provides feedback, and logs out. The system interacts with a DataPrep object to preprocess the patient data and extract relevant features, and with a MachineLearningModel object to train and predict depression based on the extracted features.

This sequence diagram shows the flow of interactions between the healthcare provider, the system, and the various objects in the system. However, there may be additional interactions depending on the specific context, users, and goals of the system.

3.6 IMPLEMENTATION

5.1 OVERVIEW OF TECHNOLOGIES

5.1.1 Python

Python is a high-level, interpreted programming language that was first released in 1991 by Guido van Rossum. It is a dynamically-typed language and is known for its simplicity and readability. Python has a wide range of applications, including web development, scientific computing, data analysis, artificial intelligence, and machine learning.

Some of the key features of Python include its easy-to-learn syntax, built-in data structures, dynamic memory management, and support for multiple programming paradigms (including procedural, object-oriented, and functional programming). Python also has a vast standard library and an active community of developers who contribute to third-party libraries and packages.

Overall, Python is a popular language for beginners and experienced developers alike, due to its ease of use, versatility, and wide range of applications.

5.1.2 Google Colab

Google Colaboratory (also known as Google Colab or simply Colab) is a free cloud-based Jupyter notebook environment that allows you to write and run Python code. It is provided by Google and runs on Google Cloud infrastructure, so you don't need to install anything on your computer to use it.

Colab provides a range of features including the ability to run code snippets, create and edit notebooks, share notebooks with others, and connect to other Google Cloud services such as Google Drive and Google Sheets. You can also use Colab to run machine learning models and experiments, as it provides access to GPUs and TPUs (Tensor Processing Units) for faster computations.

One of the main advantages of Colab is that it allows you to collaborate with others in real-time. You can share notebooks with other users and work on them together, and you can also comment on each other's code to provide feedback and suggestions.

Overall, Google Colaboratory is a powerful tool for data scientists, researchers, and students who want to experiment with Python code and machine learning models in a collaborative and cloud-based environment.

5.1.3 Google Drive

Google Drive is a cloud-based storage service provided by Google that allows you to store, share, and access files and documents from anywhere with an internet connection. With Google Drive, you can store a wide variety of file types, including documents, spreadsheets, presentations, images, and videos.

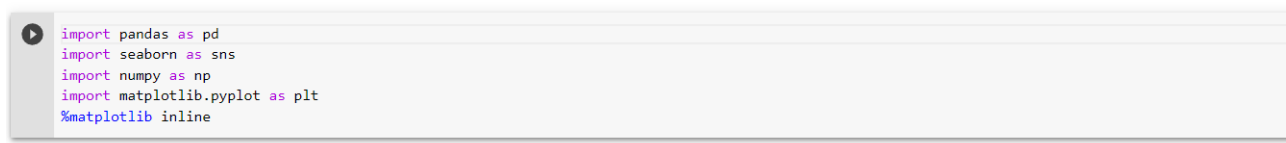
Google Drive offers a range of features, including the ability to create and edit files online using Google Docs, Sheets, and Slides, and to collaborate with others in real-time on the same document. You can also share files and folders with others, set permissions to control who can view or edit files, and access your files from any device with an internet connection.

In addition to its storage and collaboration features, Google Drive also offers integrations with other Google services such as Gmail and Google Photos, as well as third-party applications such as Adobe and DocuSign.

Overall, Google Drive is a powerful and versatile tool for individuals and teams who need to store and share files and collaborate on documents and projects. It offers a range of features and integrations that make it easy to access and work with your files from anywhere, on any device.

5.2 WORKFLOW

5.2.1 Importing Libraries



```
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

Fig 10

Pandas- Pandas is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data.

Seaborn- Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

Numpy- NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices.

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

5.2.1 Importing Dataset

```
] from google.colab import drive
drive.mount('/content/drive')

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

] df=pd.read_csv("/content/drive/MyDrive/Colab Notebooks/Depression (Responses) - Form Responses 1.csv")
```

Fig 11

df.head()

	Timestamp	Gender	Are you physically active? Do you exercise often?	Do you smoke?	Do you drink?	Do you eat healthy?	Do you have a good relationship with your parents?	Do you often feel stressed at work/college?	Are you in a relationship?	Do you tend to feel alone even if you have a lot of friends?	Do you often feel like a disappointment to people	Do you think you are depressed	Name
0	1/31/2023 14:29:50	Female	Sometimes	No	yes	I eat outside sometimes	Yes, I talk to them everyday	I get tired after work/college but I try to st...	Yes	No	No	0	NaN
1	1/31/2023 19:44:26	Female	Yes, everyday	No	no	I eat outside sometimes	Yes, I talk to them everyday	I get tired after work/college but I try to st...	No	Yes	Yes	0	Vora Sreeja
2	1/31/2023 19:45:15	Female	No	No	no	I eat outside sometimes	I talk to them sometimes or only when I need s...	I get tired after work/college but I try to st...	No	Yes	Yes	0	M.G Manjusha
3	1/31/2023 19:45:17	Female	Sometimes	No	no	I eat outside sometimes	Yes, I talk to them everyday	I get tired after work/college but I try to st...	No	Yes	Yes	0	Kavya Reddy
4	1/31/2023 19:46:13	Female	No	No	no	I eat outside sometimes	Yes, I talk to them everyday	I get so tired that I can't even move after wo...	No	Yes	Yes	1	Divya

Fig 12

```
df.isnull().sum()

Timestamp      0
Gender          0
Are you physically active? Do you exercise often?    0
Do you smoke?    0
Do you drink?    0
Do you eat healthy?    0
Do you have a good relationship with your parents?    0
Do you often feel stressed at work/college?    0
Are you in a relationship?    0
Do you tend to feel alone even if you have a lot of friends?    0
Do you often feel like a disappointment to people    0
Do you think you are depressed    0
Name           2
dtype: int64
```

wo people did not fill their names. Except that there are no null values

Fig 13

From the above images, we can see all the questions that we asked in order to create the dataset and also see the first 5 rows of the dataset.

5.2.3 Exploratory Data Analysis

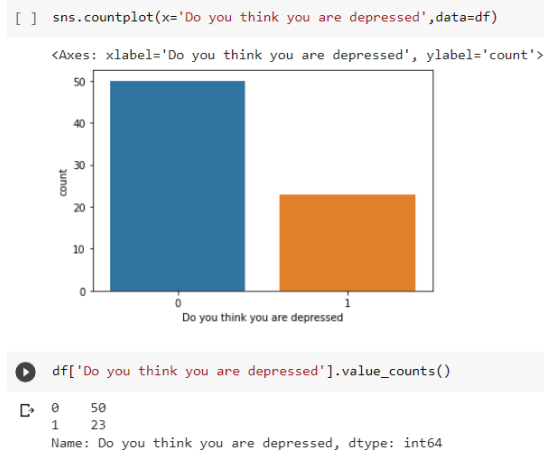


Fig 14

From the above graph we can observe that out of the 73 people who filled out filled, 50 people are not depressed whereas 23 people are depressed.



Fig 15

In the above graph we compared the gender of the person with if they are depressed or not and we see that Gender does not have a huge impact on depression.

```
[ ] sns.countplot(x='Do you think you are depressed',hue='Are you physically active? Do you exercise often?',data=df)
```

<Axes: xlabel='Do you think you are depressed', ylabel='count'>

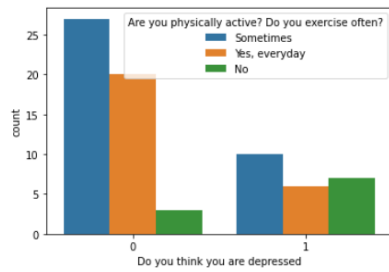


Fig 16

In the above graph we compared to see if people who are physically active with their depression state and we see that, people who are not depressed work out every day or even occasionally.

```
[ ] sns.countplot(x='Do you think you are depressed',hue='Do you smoke?',data=df)
```

<Axes: xlabel='Do you think you are depressed', ylabel='count'>

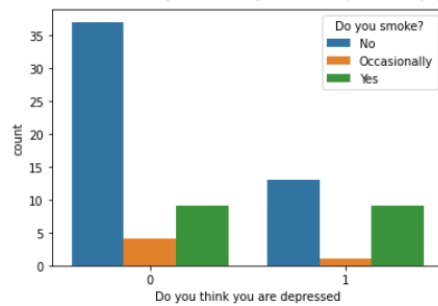


Fig 17

In the above graph, we compared to see if people who smoke are depressed and we observed that people who are not depressed usually do not smoke at all.

```
▶ sns.countplot(x='Do you think you are depressed',hue='Do you drink?',data=df)
```

☞ <Axes: xlabel='Do you think you are depressed', ylabel='count'>

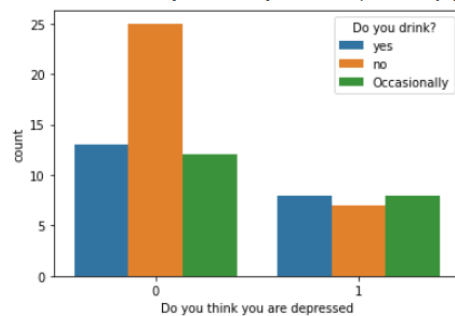


Fig 18

In the above graph, we compared to see if drinking affects the persons' depressed state and we see that people who don't drink at all are usually not depressed.

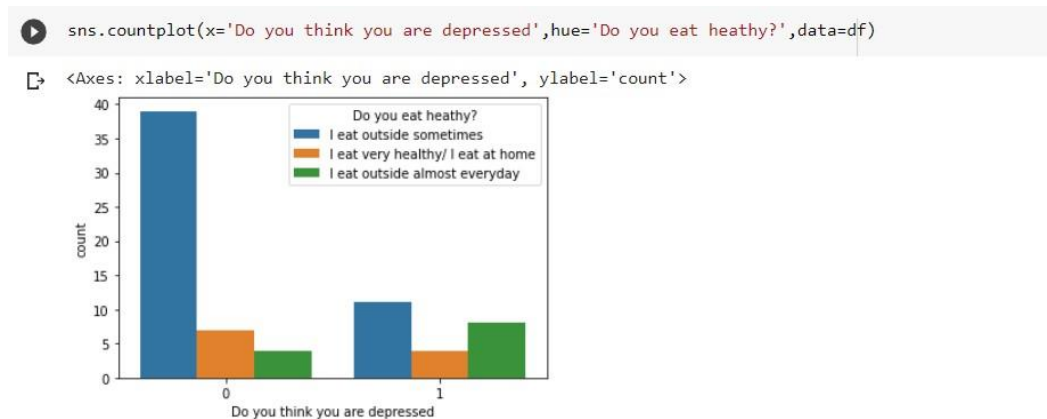


Fig 19

In the above graph, we compared to see if eating habits affected depression and we see that people who are not depressed eat out sometimes.

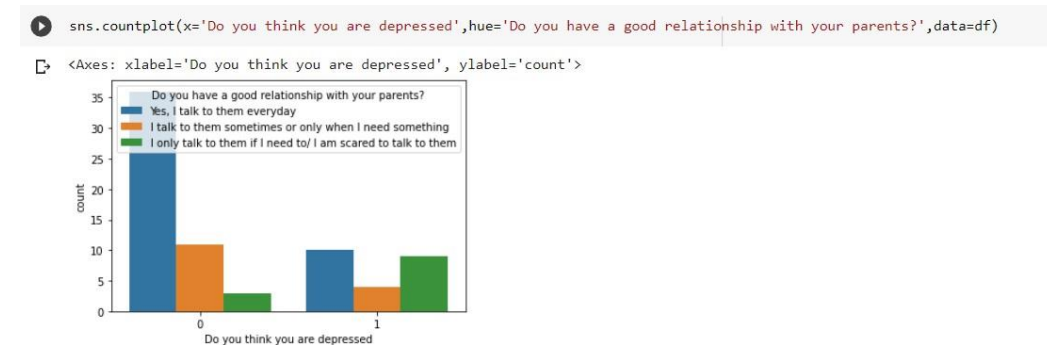


Fig 20

In the above graph, we compared to see if their relationship with their parents affected depression and we see that More people who talk to their parents everyday are not depressed and people who talk to their parents only when they need something seem more depressed.

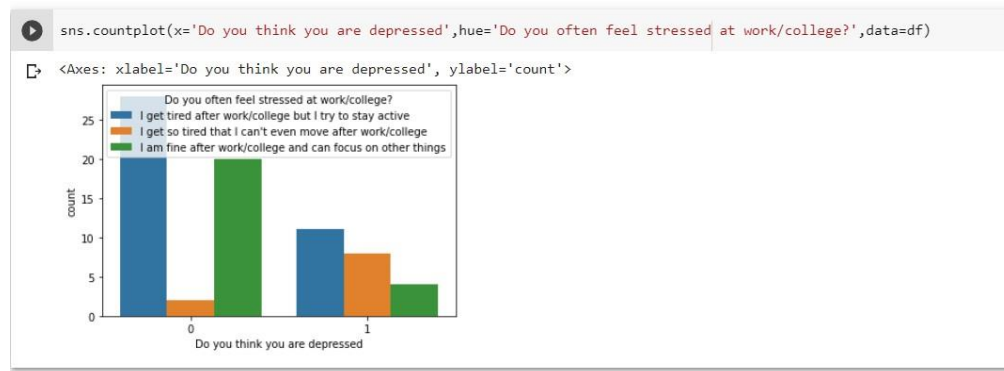


Fig 21

In the above graph, we tried to see if people feeling stressed is related to depression and we see that people who are not depressed seem to be active even after work and people who are depressed can't even move

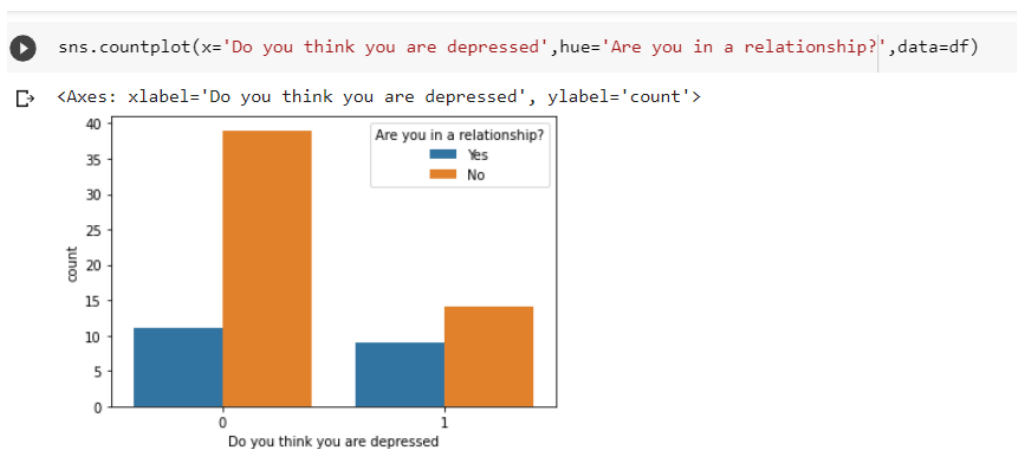


Fig 22

In the above graph, we tried to see if relationship status affects depression and we see that, people who are not in a relationship are usually not depression.

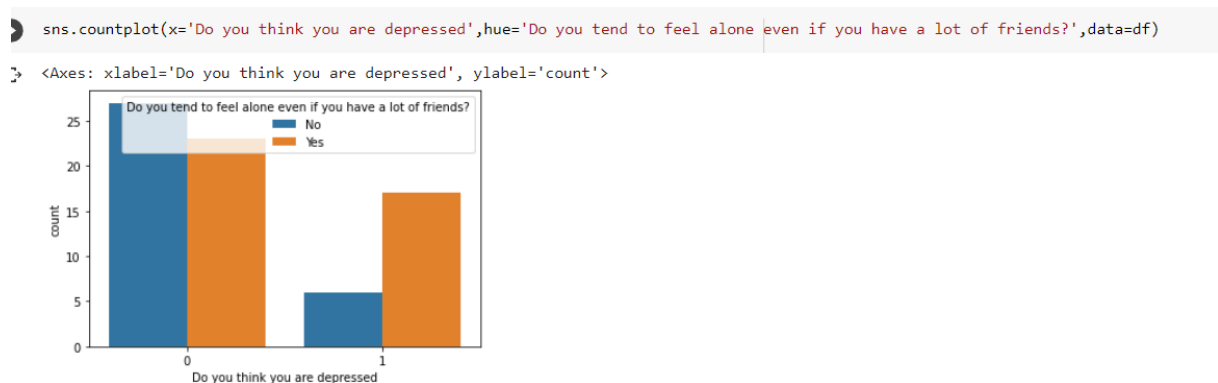


Fig 23

In the above graph, we tried to see if people feel alone when depressed and we see that, people who are not depressed usually do not feel alone eve if they have a lot of friends.

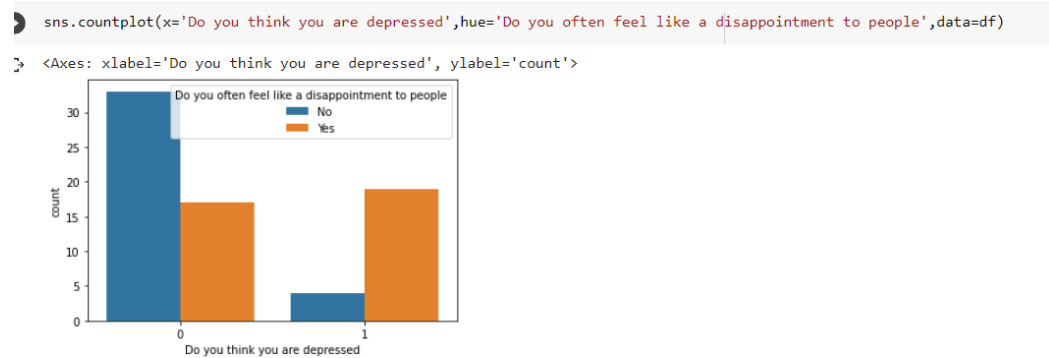


Fig 24

In the above graph, we tried to see if people feel like a disappointment when depressed and we see that most not depressed people do not feel like disappointment.

5.2.4 Data Preprocessing

We collected 20 more responses for the dataset and combined both the existing data and the new data for easier preprocessing

```
test= pd.read_csv("/content/drive/MyDrive/Colab Notebooks/Depression-test (Responses) - Form Responses 1.csv")
combine = [df,test]
df.head()
```

	Timestamp	Gender	Are you physically active? Do you exercise often?	Do you smoke?	Do you drink?	Do you eat heathy?	Do you have a good relationship with your parents?	Do you often feel stressed at work/college?	Are you in a relationship?	Do you tend to feel alone even if you have a lot of friends?	Do you often feel like a disappointment to people	Do you think you are depressed	Name
0	1/31/2023 14:29:50	Female	Sometimes	No	yes	I eat outside sometimes	Yes, I talk to them everyday	I get tired after work/college but I try to st...	Yes	No	No	0	NaN
1	1/31/2023 19:44:26	Female	Yes, everyday	No	no	I eat outside sometimes	Yes, I talk to them everyday	I get tired after work/college but I try to st...	No	Yes	Yes	0	Vora Sreeja
2	1/31/2023 19:45:15	Female	No	No	no	I eat outside sometimes	I talk to them sometimes or only when I need s...	I get tired after work/college but I try to st...	No	Yes	Yes	0	M.G.Manjusha
3	1/31/2023 19:45:17	Female	Sometimes	No	no	I eat outside sometimes	Yes, I talk to them everyday	I get tired after work/college but I try to st...	No	Yes	Yes	0	Kavya Reddy
4	1/31/2023 19:46:13	Female	No	No	no	I eat outside sometimes	Yes, I talk to them everyday	I get so tired that I can't even move after wo...	No	Yes	Yes	1	Divya

Fig 25

We started by dropping the unnecessary columns like ‘Timestamp’ and ‘Name’

```
[ ] df.drop(['Timestamp','Name'],axis=1,inplace=True)
```

```
df.head()
```

	Gender	Are you physically active? Do you exercise often?	Do you smoke?	Do you drink?	Do you eat healthy?	Do you have a good relationship with your parents?	Do you often feel stressed at work/college?	Are you in a relationship?	Do you tend to feel alone even if you have a lot of friends?	Do you often feel like a disappointment to people	Do you think you are depressed
0	Female	Sometimes	No	yes	I eat outside sometimes	Yes, I talk to them everyday	I get tired after work/college but I try to st...	Yes	No	No	0
1	Female	Yes, everyday	No	no	I eat outside sometimes	Yes, I talk to them everyday	I get tired after work/college but I try to st...	No	Yes	Yes	0
2	Female	No	No	no	I eat outside sometimes	I talk to them sometimes or only when I need s...	I get tired after work/college but I try to st...	No	Yes	Yes	0
3	Female	Sometimes	No	no	I eat outside sometimes	Yes, I talk to them everyday	I get tired after work/college but I try to st...	No	Yes	Yes	0
4	Female	No	No	no	I eat outside sometimes	Yes, I talk to them everyday	I get so tired that I can't even move after wo...	No	Yes	Yes	1

Fig 26

Then in order to fit the values into the Machine Learning Models, we need to convert the values into Categorical Values.

```
[ ] pd.get_dummies(df['Gender'],drop_first=True).head()
```

	Male
0	0
1	0
2	0
3	0
4	0

```
pd.get_dummies(df['Are you physically active? Do you exercise often?'],drop_first=True).head()
```

	Sometimes	Yes, everyday
0	1	0
1	0	1
2	0	0
3	1	0
4	0	0

Fig 27

```
] pd.get_dummies(df['Do you smoke?'],drop_first=True).head()
```

	Occasionally	Yes
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0

```
] pd.get_dummies(df['Do you drink?'],drop_first=True).head()
```

	no	yes
0	0	1
1	1	0
2	1	0
3	1	0
4	1	0

Fig 28

```
pd.get_dummies(df['Do you eat heathy?'],drop_first=True).head()
```

	I eat outside sometimes	I eat very healthy/ I eat at home
0	1	0
1	1	0
2	1	0
3	1	0
4	1	0

```
pd.get_dummies(df['Do you have a good relationship with your parents?'],drop_first=True).head()
```

	I talk to them sometimes or only when I need something	Yes, I talk to them everyday
0	0	1
1	0	1
2	1	0
3	0	1
4	0	1

Fig 29

```
] pd.get_dummies(df['Do you often feel stressed at work/college?'],drop_first=True).head()
```

	I get so tired that I can't even move after work/college	I get tired after work/college but I try to stay active
0	0	1
1	0	1
2	0	1
3	0	1
4	1	0

```
] pd.get_dummies(df['Do you tend to feel alone even if you have a lot of friends?'],drop_first=True).head()
```

	Yes
0	0
1	1
2	1
3	1
4	1

Fig 30

```
] pd.get_dummies(df['Do you often feel like a disappointment to people'],drop_first=True).head()
```

	Yes
0	0
1	1
2	1
3	1
4	1

```
pd.get_dummies(df['Are you in a relationship?'],drop_first=True).head()
```

	Yes
0	1
1	0
2	0
3	0
4	0

Fig 31

In the above diagrams, we can see that we converted all the columns into categorical values and now we need to append these columns to the dataset and get rid of the existing columns.

```
[ ] gendr=pd.get_dummies(df['Gender'],drop_first=True).head()
smoke=pd.get_dummies(df['Do you smoke?'],drop_first=True).head()
drink=pd.get_dummies(df['Do you drink?'],drop_first=True).head()
active=pd.get_dummies(df['Are you physically active? Do you exercise often?'],drop_first=True)
healthy=pd.get_dummies(df['Do you eat healthy?'],drop_first=True)
parents=pd.get_dummies(df['Do you have a good relationship with your parents?'],drop_first=True)
stress=pd.get_dummies(df['Do you often feel stressed at work/college?'],drop_first=True)
alone=pd.get_dummies(df['Do you tend to feel alone even if you have a lot of friends?'],drop_first=True)
disappointment=pd.get_dummies(df['Do you often feel like a disappointment to people?'],drop_first=True)
relation=pd.get_dummies(df['Are you in a relationship?'],drop_first=True).head()

[ ] df.drop(['Gender','Do you smoke?','Do you drink?','Are you in a relationship?','Are you physically active? Do you exercise often?','Do you eat healthy?','Do you have a good relationship with your parents?'],
df.head()
```

	Do you think you are depressed
0	0
1	0
2	0
3	0
4	1

Fig 32

We create new columns and then drop all the existing columns in the dataset.

```
[ ] df = pd.concat([df, gendr,smoke,drink,active,healthy,parents,stress,alone,disappointment,relation],axis=1,join='inner')
df.head()
```

	Do you think you are depressed	Male	Occasionally	Yes	no	yes	Sometimes	Yes, everyday	I eat outside sometimes	I eat very healthy/ I eat at home	I talk to them sometimes or only when I need something	Yes, I talk to them everyday	I get so tired that I can't even move after work/college	I get tired after work/college but I try to stay active	Yes	Yes	Yes
0	0	0	0	0	0	1	1	0	1	0	0	1	0	1	0	0	1
1	0	0	0	0	1	0	0	1	1	0	0	1	0	1	1	1	0
2	0	0	0	0	1	0	0	0	1	0	1	0	0	1	1	1	0
3	0	0	0	0	1	0	1	0	1	0	0	1	0	1	1	1	0
4	1	0	0	0	1	0	0	0	1	0	0	1	1	0	1	1	0

Fig 33

We concatenated the columns that we made to the dataset so now all the values are in form of 0s and 1s.

```
df.drop("Do you think you are depressed",axis=1).head()
```

	Hale	Occasionally	Yes	no	yes	Sometimes	Yes, everyday	I eat outside sometimes	I eat very healthy/ I eat at home	I talk to them sometimes or only when I need something	Yes, I talk to them everyday	I get so tired that I can't even move after work/college	I get tired after work/college but I try to stay active	Yes	Yes	Yes
0	0	0	0	0	1	1	0	1	0	0	1	0	1	0	0	1
1	0	0	0	1	0	0	1	1	0	0	1	0	0	1	1	1
2	0	0	0	1	0	0	0	1	0	1	0	0	0	1	1	1
3	0	0	0	1	0	1	0	1	0	0	1	0	0	1	1	1
4	0	0	0	1	0	0	0	1	0	0	1	1	0	0	1	1

```
df["Do you think you are depressed"].head()
```

```
0    0
1    0
2    0
3    0
4    1
Name: Do you think you are depressed, dtype: int64
```

Fig 34

We remove the depressed column as that is the value we aim to predict.

5.2.5 Spitting dataset into training and test set

```
X_train = df.drop("Do you think you are depressed", axis=1)
y_train = df["Do you think you are depressed"]

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(df.drop('Do you think you are depressed',axis=1),
                                                    df['Do you think you are depressed'], test_size=0.40,
                                                    random_state=101)
```

Fig 35

We use the train-test split technique in machine learning to evaluate the performance of a model on an independent dataset. The idea is to split the available data into two sets: one for training the model and the other for testing its performance.

The training dataset is used to fit the model, while the testing dataset is used to evaluate its generalization performance. The idea is to use the testing dataset to estimate how well the model will perform on new, unseen data.

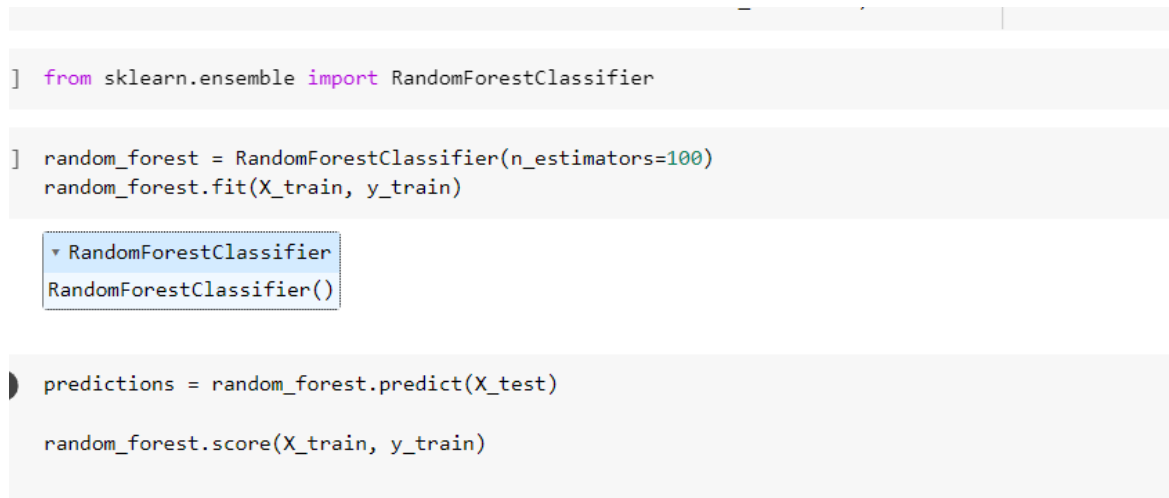
If we do not split the data and use the same dataset for both training and testing, the model can become overfitted to the training data, meaning that it performs well on the training data but poorly on the testing data. This is because the model has learned the idiosyncrasies and noise of the training data, which do not generalize well to new data.

By using a separate testing dataset, we can estimate the true performance of the model on new, unseen data. This allows us to select the best model among several candidate models or to tune the hyperparameters of a model to improve its performance.

The train-test split is a simple and effective technique for evaluating the performance of a model, but it has some limitations.

5.2.6 Applying the Machine Learning Models

RANDOM FOREST



```
] from sklearn.ensemble import RandomForestClassifier

random_forest = RandomForestClassifier(n_estimators=100)
random_forest.fit(X_train, y_train)

▼ RandomForestClassifier
RandomForestClassifier()

predictions = random_forest.predict(X_test)

random_forest.score(X_train, y_train)
```

Fig 36

We import the Random Forest Classifier from the sklearn library and apply it onto our dataset and then we check for accuracy.

```

] from sklearn.metrics import confusion_matrix

] accuracy=confusion_matrix(y_test,predictions)

] accuracy

array([[1, 0],
       [1, 0]])

] from sklearn.metrics import accuracy_score

accuracy=accuracy_score(y_test,predictions)
accuracy

0.5

```

Fig 37

From the above figure, we can see that we obtained 50% accuracy by using Random Forest Classifier.

GAUSSIAN NAÏVE BAYES

```

] from sklearn.naive_bayes import GaussianNB

gnb = GaussianNB()

gnb.fit(X_train, y_train)

GaussianNB
GaussianNB()

] y_pred = gnb.predict(X_test)

y_pred

array([0, 0])

```

Fig 38

We import the Gaussian Naïve Bayes from the sklearn library and apply it onto our dataset and then we check for accuracy.

```
[ ] accuracy1=confusion_matrix(y_test,y_pred)

[ ] accuracy1

array([[1, 0],
       [1, 0]])

[ ] accuracy1=accuracy_score(y_test,y_pred)
accuracy1

0.5
```

Fig 39

From the above figure, we can see that we obtained 50% accuracy by using Gaussian Naïve Bayes.

DECISION TREE CLASSIFIER

```
[ ] from sklearn.tree import DecisionTreeClassifier
    clf = DecisionTreeClassifier()
    clf = clf.fit(X_train,y_train)

▶ y_pred2 = clf.predict(X_test)

▶ y_pred2

array([0, 0])
```

Fig 40

We import the Decision Tree Classifier from the sklearn library and apply it onto our dataset and then we check for accuracy.

```
[ ] accuracy2=confusion_matrix(y_test,y_pred)
accuracy2

array([[1, 0],
       [1, 0]])

[ ] accuracy2=accuracy_score(y_test,y_pred)
accuracy2

0.5
```

Fig 41

From the above figure, we can see that we obtained 50% accuracy by using Decision Tree Classifier.

K Nearest Neighbors

```
[ ] from sklearn.neighbors import KNeighborsClassifier
    knn = KNeighborsClassifier(n_neighbors = 2)
    knn.fit(X_train, y_train)
    Y_pred = knn.predict(X_test)
```

Fig 42

We import the K Nearest Neighbors from the sklearn library and apply it onto our dataset and then we check for accuracy.

```
] accuracy3=confusion_matrix(y_test,Y_pred)
accuracy3

array([[1, 0],
       [1, 0]])

] accuracy3=accuracy_score(y_test,Y_pred)
accuracy3

0.5
```

Fig 43

From the above figure, we can see that we obtained 50% accuracy by using K Nearest Neighbors.

MLP CLASSIFIER

```
] from sklearn.neural_network import MLPClassifier
    mlp = MLPClassifier()
    mlp.fit(X_train,y_train)
    Y_pred = mlp.predict(X_test)
    mlp.score(X_train, y_train)

1.0
```

Fig 44

We import the MLP Classifier from the sklearn library and apply it onto our dataset and then we check for accuracy.


```
] accuracy5=confusion_matrix(y_test,Y_pred)
accuracy5

array([[1, 0],
       [1, 0]])

] accuracy5=accuracy_score(y_test,Y_pred)
accuracy5

0.5
```

Fig 45

From the above figure, we can see that we obtained 50% accuracy by using MLP Classifier.

5.3 LIBRARIES IMPORTED

5.3.1 Pandas

Pandas is a Python library designed for data manipulation and analysis. It provides easy-to-use data structures, such as Series and DataFrame, to represent and manipulate tabular data. Pandas allows importing data from various sources such as CSV, Excel, SQL databases, and web APIs. It supports data cleaning, filtering, aggregation, grouping, and merging operations. Pandas can handle missing values, time series data, and text data. It also provides advanced data visualization functions using Matplotlib and Seaborn.

5.3.2 Seaborn

Seaborn is one of an amazing library for visualization of the graphical statistical plotting in Python. Seaborn provides many color palettes and defaults beautiful styles to make the creation of many statistical plots in Python more attractive. eaborn library aims to make a more attractive visualization of the central part of understanding and exploring data. It is built on the core of the matplotlib library and also provides dataset-oriented APIs.

Seaborn is also closely integrated with the Panda's data structures, and with this, we can easily jump between the various different visual representations for a given variable to better understand the provided dataset.

5.3.3 Numpy

NumPy is a Python library designed for numerical computation and scientific computing. It provides powerful data structures, such as arrays and matrices, for representing and manipulating numerical data. NumPy supports vectorized operations and mathematical functions, making it faster and more efficient than regular Python lists. It also provides advanced mathematical and statistical functions, such as linear algebra, Fourier transforms, and

random number generation. NumPy integrates well with other libraries in the scientific Python ecosystem, such as Pandas, SciPy, and Matplotlib.

5.3.4 Matplotlib

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002. One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc. Installation : Windows, Linux and macOS distributions have matplotlib and most of its dependencies as wheel packages

5.3.5 Sklearn

Scikit-learn, also known as sklearn, is a Python library used for machine learning and data analysis. It provides a variety of tools for supervised and unsupervised learning, including classification, regression, clustering, and dimensionality reduction. Scikit-learn is built on top of other scientific Python libraries such as NumPy, Pandas, and Matplotlib, and provides a consistent interface to the machine learning algorithms. Scikit-learn supports a wide range of machine learning models, including Naive Bayes, Decision Trees, Random Forests, SVM, KNN, Neural Networks, and many others. It also provides tools for model evaluation, hyperparameter tuning, and model selection.

5.4 DATASET

5.4.1 Data Collection

We sent a google form with a few questions to everyone in our circle, with the following questions.

- Gender
- Are you physically active? Do you exercise often?
- Do you smoke?
- Do you drink?
- Do you eat healthy?
- Do you have a good relationship with your parents?
- Do you often feel stressed at work/college?
- Are you in a relationship?
- Do you tend to feel alone even if you have a lot of friends?
- Do you often feel like a disappointment to people
- Do you think you are depressed

Depression Survey

Please fill out the questionnaire. Your data will not be shared with anyone except the owners of the form so please be honest with your answers

Name

Short answer text

Gender

☐ Male
☐ Female

Are you physically active? Do you exercise often?

☐ Yes, everyday
☐ Sometimes
☐ No

+

Tr

Fig 46

Do you drink?

☐ yes
☐ no
☐ Occasionally

Do you eat healthy?

☐ I eat very healthy/ I eat at home
☐ I eat outside sometimes
☐ I eat outside almost everyday

Do you have a good relationship with your parents?

☐ Yes, I talk to them everyday
☐ I talk to them sometimes or only when I need something
☐ I only talk to them if I need to/ I am scared to talk to them

+

Tr

Fig 47

Do you often feel stressed at work/college?
☐ I am fine after work/college and can focus on other things
☐ I get tired after work/college but I try to stay active
☐ I get so tired that I can't even move after work/college

Are you in a relationship?
☐ Yes
☐ No

Do you tend to feel alone even if you have a lot of friends?
☐ Yes
☐ No

Do you often feel like a disappointment to people
☐ Yes
☐ No

+
-
Tr
+
-

Fig 48

Do you often feel like a disappointment to people
☐ Yes
☐ No

Do you think you are depressed
☐ Yes
☐ No

Fig 49

5.4.2 Final Data

We converted the data collected into an Excel sheet which we downloaded in the form of csv file for our use.

	A	B	C	D	E	F	G	H	I	J	K	L	
1	Timestamp	Gender	Are you physically active	Do you smoke?	Do you drink?	Do you eat healthy?	Do you have a good relationship?	Do you often feel stressed at work/college?	Are you in a relationship?	Do you tend to feel alone even if you have a lot of friends?	Do you often feel like a disappointment to people?	Do you think you are depressed?	Name
2	1/31/2023 14:29:50	Female	Sometimes	No	yes	I eat outside sometimes	Yes, I talk to them everyc	I get tired after work/college	Yes	No	No	No	0
3	1/31/2023 19:44:26	Female	Yes, everyday	No	no	I eat outside sometimes	Yes, I talk to them everyc	I get tired after work/college	No	Yes	Yes	Yes	0
4	1/31/2023 19:45:15	Female	No	No	no	I eat outside sometimes	I talk to them sometimes	I get tired after work/college	No	Yes	Yes	Yes	0
5	1/31/2023 19:45:17	Female	Sometimes	No	no	I eat outside sometimes	Yes, I talk to them everyc	I get tired after work/college	No	Yes	Yes	Yes	0
6	1/31/2023 19:46:13	Female	No	No	no	I eat outside sometimes	Yes, I talk to them everyc	I get so tired that I can't e	No	Yes	Yes	Yes	1
7	1/31/2023 19:50:15	Female	Sometimes	No	yes	I eat very healthy/ I eat a	Yes, I talk to them everyc	I get tired after work/college	No	No	No	No	0
8	1/31/2023 19:52:01	Female	Sometimes	No	Occasionally	I eat outside sometimes	Yes, I talk to them everyc	I get tired after work/college	Yes	Yes	Yes	Yes	1
9	1/31/2023 19:52:06	Female	Sometimes	No	no	I eat outside sometimes	Yes, I talk to them everyc	I get tired after work/college	No	Yes	Yes	Yes	0
10	1/31/2023 19:54:31	Male	Sometimes	No	no	I eat outside sometimes	I talk to them sometimes	I get tired after work/college	No	No	No	No	0
11	1/31/2023 19:55:30	Male	No	No	no	I eat very healthy/ I eat a	I talk to them sometimes	I get tired after work/college	No	Yes	Yes	Yes	1
12	1/31/2023 19:58:01	Female	Yes, everyday	No	no	I eat outside sometimes	Yes, I talk to them everyc	I get tired after work/college	No	Yes	No	No	0
13	1/31/2023 19:59:14	Female	Sometimes	No	no	I eat outside sometimes	I talk to them sometimes	I get tired after work/college	No	Yes	Yes	Yes	0
14	1/31/2023 20:04:01	Female	Sometimes	No	no	I eat outside sometimes	Yes, I talk to them everyc	I am fine after work/college	Yes	No	Yes	Yes	1
15	1/31/2023 20:12:26	Male	No	Occasionally	Occasionally	I eat outside almost ever	I only talk to them if I nee	I get so tired that I can't e	No	Yes	Yes	Yes	1
16	1/31/2023 20:13:00	Female	Sometimes	Yes	Occasionally	I eat outside sometimes	Yes, I talk to them everyc	I get tired after work/college	Yes	Yes	No	No	1
17	1/31/2023 20:27:33	Male	Sometimes	No	no	I eat outside almost ever	Yes, I talk to them everyc	I get tired after work/college	No	Yes	Yes	Yes	0
18	1/31/2023 20:27:52	Male	Sometimes	No	Occasionally	I eat outside sometimes	Yes, I talk to them everyc	I am fine after work/college	No	No	No	No	0
19	1/31/2023 20:41:20	Male	No	No	no	I eat outside sometimes	I talk to them sometimes	I get so tired that I can't e	No	Yes	Yes	Yes	1
20	1/31/2023 20:54:11	Female	Sometimes	No	yes	I eat outside sometimes	Yes, I talk to them everyc	I get tired after work/college	Yes	Yes	Yes	Yes	0
21	1/31/2023 21:13:55	Male	Yes, everyday	No	no	I eat outside sometimes	Yes, I talk to them everyc	I am fine after work/college	No	No	No	No	0
22	1/31/2023 21:44:40	Female	Sometimes	No	yes	I eat outside almost ever	I only talk to them if I nee	I get so tired that I can't e	Yes	Yes	Yes	Yes	0
23	1/31/2023 21:45:52	Female	Sometimes	No	Occasionally	I eat outside sometimes	I only talk to them if I nee	I get so tired that I can't e	No	Yes	Yes	Yes	1
24	1/31/2023 21:48:17	Male	Sometimes	No	Occasionally	I eat outside sometimes	Yes, I talk to them everyc	I get tired after work/college	Yes	No	No	No	0
25	1/31/2023 21:50:01	Male	Yes, everyday	Occasionally	no	I eat outside sometimes	Yes, I talk to them everyc	I get tired after work/college	No	No	No	No	0
26	1/31/2023 21:50:16	Male	Yes, everyday	No	no	I eat outside sometimes	Yes, I talk to them everyc	I am fine after work/college	Yes	Yes	No	No	0
27	1/31/2023 21:52:16	Male	Yes, everyday	Yes	Occasionally	I eat outside sometimes	Yes, I talk to them everyc	I am fine after work/college	No	No	No	No	0
28	1/31/2023 21:52:17	Female	Sometimes	No	no	I eat very healthy/ I eat a	I talk to them sometimes	I am fine after work/college	No	No	Yes	Yes	0
29	1/31/2023 21:54:03	Female	Sometimes	Yes	yes	I eat outside almost ever	I only talk to them if I nee	I get so tired that I can't e	Yes	Yes	Yes	Yes	1
30	1/31/2023 21:54:23	Female	Yes, everyday	No	Occasionally	I eat outside sometimes	Yes, I talk to them everyc	I am fine after work/college	No	No	No	No	0
31	1/31/2023 21:55:07	Female	Sometimes	No	Occasionally	I eat outside sometimes	I only talk to them if I nee	I get tired after work/college	No	Yes	Yes	Yes	0
32	1/31/2023 21:58:32	Female	No	No	Occasionally	I eat outside almost ever	Yes, I talk to them everyc	I get tired after work/college	No	Yes	No	No	1

Fig 50

The above picture that is Fig 50 contains the data.

5.5 ALGORITHMS IMPLEMENTED

5.5.1 Random Forest

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

5.5.2 Gaussian Naïve Bayes

Gaussian Naive Bayes is a probabilistic algorithm that uses Bayes' theorem to classify data. It assumes that the features are independent and follow a Gaussian distribution. The algorithm calculates the probability of each class given the input data and chooses the class with the highest probability as the output. It is commonly used for classification tasks such as spam filtering, sentiment analysis, and text classification.

5.5.3 Decision Tree

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. The decisions or the test are performed on the basis of features of the given dataset. It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions. It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure. In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm. A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.

5.5.4 K Nearest Neighbors

K-Nearest Neighbors (KNN) is a non-parametric machine learning algorithm used for classification and regression tasks. It works by finding the K closest data points in the feature space to a new instance and assigning

the label or value of the majority of these neighbors to the new instance. The value of K is a hyperparameter that needs to be tuned to balance the bias-variance tradeoff. KNN is a lazy learning algorithm that does not require any training time but can be slow at prediction time for large datasets.

5.5.5 MLP Classifier

Multilayer Perceptron (MLP) Classifier is a neural network-based machine learning algorithm used for classification tasks. It consists of multiple layers of nodes, including input, hidden, and output layers. Each node in the hidden and output layers uses an activation function to transform the input signal and produce an output. The algorithm learns the weights and biases of the connections between the layers by minimizing a loss function during training. MLP can capture complex non-linear relationships between features and can handle both categorical and numerical data.

3.7 TESTING AND VALIDATION

6.1 SYSTEM TESTING

System testing is an essential part of software development and is conducted to ensure that the entire system functions as intended and meets the specified requirements.

6.1.1 Accuracy Testing for Random Forest

Accuracy is a metric commonly used in machine learning to evaluate the performance of a classification model. It measures the proportion of correct predictions made by the model over the total number of predictions.

The accuracy can be calculated using the following formula:

$$\text{Accuracy} = (\text{Number of Correct Predictions}) / (\text{Total Number of Predictions})$$

For example, if a model correctly predicts 80 out of 100 instances, its accuracy is 80%.

While accuracy is a simple and intuitive metric, it can be misleading in certain situations. For instance, in a highly imbalanced dataset, where one class is much more frequent than the other, the model can achieve high accuracy by simply predicting the majority class every time. In such cases, other metrics such as precision, recall, and F1-score should be used to evaluate the model's performance.

Also, accuracy alone may not be sufficient to evaluate the performance of a model in some applications. For instance, in medical diagnosis, false negatives (failing to identify a disease when it is present) can be more

severe than false positives (identifying a disease when it is not present). In such cases, a metric such as sensitivity (the proportion of true positives among all positive cases) or specificity (the proportion of true negatives among all negative cases) can be more informative.

Therefore, while accuracy is a useful metric, it should be interpreted with caution and used in conjunction with other metrics to evaluate the performance of a model in a specific context.

In the case of our project,

```
[29] from sklearn.metrics import accuracy_score

[30] accuracy=accuracy_score(y_test,predictions)
      accuracy

0.5
```

Fig 51

By using Random Forest Classifier, we can see that we obtained 50% accuracy

6.1.2 Accuracy Testing for Gaussian Naïve Bayes

```
[37] accuracy1=accuracy_score(y_test,y_pred)
      accuracy1

0.5
```

Fig 52

By using Gaussian Naïve Bayes, we can see that we obtained 50% accuracy

6.1.3 Accuracy Testing for Decision Tree Classifier

```
43] accuracy2=accuracy_score(y_test,y_pred2)
      accuracy2

0.5
```

Fig 53

By using Decision Tree Classifier, we can see that we obtained 50% accuracy.

6.1.4 Accuracy Testing for K Nearest Neighbors

```
46] accuracy3=accuracy_score(y_test,y_pred2)
accuracy3

0.5
```

Fig 54

By using K Nearest Neighbors, we can see that we obtained 50% accuracy.

6.1.5 Accuracy Testing for MLP Classifier

```
[50] accuracy5=accuracy_score(y_test,y_pred3)
accuracy5

0.5
```

Fig 55

By using MLP Classifier, we can see that we obtained 50% accuracy

6.2 PERFORMANCE METRICS FOR THE MODELS

Precision, in the context of measurement and statistics, refers to the degree of accuracy or exactness of a measurement or calculation. It is a measure of how closely individual measurements or calculations agree with each other. In statistical analysis, precision is typically measured by calculating the standard deviation of a set of measurements or values. A lower standard deviation indicates that the measurements are more precise and have less variability. In practical terms, precision is important in many fields such as engineering, science, and medicine, where accurate and precise measurements are essential. For example, in manufacturing, precision is critical to ensure that products are consistent and meet quality standards. In scientific research, precision is essential to ensure that experimental results are accurate and reproducible. In summary, precision refers to the level of accuracy or exactness of a measurement or calculation, and is an important consideration in many fields where accuracy is essential.

Recall is a measure of a model's ability to identify all relevant instances or cases within a dataset. It is typically used in the context of machine learning and information retrieval, where it is important to find all relevant instances of a particular class or category. Recall is calculated as the number of true positives (i.e., the number of

relevant instances that were correctly identified) divided by the total number of actual positive instances (i.e., the sum of true positives and false negatives). A high recall score indicates that the model is good at identifying all relevant instances, while a low recall score indicates that the model is missing some of the relevant instances. In practical terms, recall is important in many applications, such as search engines, fraud detection, and medical diagnosis. For example, in a medical diagnosis system, a high recall score would mean that the system is able to identify all patients with a particular disease, while a low recall score would mean that some patients with the disease would be missed. In summary, recall is a measure of a model's ability to identify all relevant instances, and is an important consideration in many applications where identifying all relevant instances is critical.

The F1 score is a measure of a model's accuracy that takes into account both precision and recall. It is the harmonic mean of precision and recall, and provides a single metric that balances both measures. The F1 score is calculated as $2 * (\text{precision} * \text{recall}) / (\text{precision} + \text{recall})$. It ranges from 0 to 1, with 1 indicating perfect precision and recall, and 0 indicating poor performance. In practical terms, the F1 score is often used to evaluate the performance of classification models, where it is important to balance both precision (the proportion of true positives among all positive predictions) and recall (the proportion of true positives that are correctly identified). The F1 score is particularly useful when the dataset is imbalanced, meaning that one class is more prevalent than another. In such cases, accuracy alone may be misleading, as a model that always predicts the most common class may achieve high accuracy, but may have poor performance in identifying the less common class. The F1 score takes into account both precision and recall, and provides a more balanced measure of the model's performance. In summary, the F1 score is a measure of a model's accuracy that takes into account both precision and recall, and is particularly useful in evaluating classification models with imbalanced datasets.

6.2.1 Performance Metrics for Random Forest

```

] print(classification_report(y_test,predictions))

              precision    recall  f1-score   support

     0       0.50      1.00      0.67         1
     1       0.00      0.00      0.00         1

 accuracy          0.50         2
 macro avg          0.25         2
 weighted avg       0.25         2

/usr/local/lib/python3.9/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `z
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.9/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `z
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.9/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `z
_warn_prf(average, modifier, msg_start, len(result))

```

Fig 56

In Fig 56, we can see the performance Metrics for the Random Forest Classifier.

6.2.2 Performance Metrics for Gaussian Naïve Bayes

```
[ ] from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
0	0.50	1.00	0.67	1
1	0.00	0.00	0.00	1
accuracy			0.50	2
macro avg	0.25	0.50	0.33	2
weighted avg	0.25	0.50	0.33	2

```
/usr/local/lib/python3.9/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `z`
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.9/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `z`
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.9/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `z`
_warn_prf(average, modifier, msg_start, len(result))
```

Fig 57

In Fig 57, we can see the performance Metrics for the Gaussian Naïve Bayes Model.

6.2.3 Performance Metrics for Decision Tree

```
[ ] print(classification_report(y_test,y_pred2))
```

	precision	recall	f1-score	support
0	0.50	1.00	0.67	1
1	0.00	0.00	0.00	1
accuracy			0.50	2
macro avg	0.25	0.50	0.33	2
weighted avg	0.25	0.50	0.33	2

```
/usr/local/lib/python3.9/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `z`
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.9/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `z`
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.9/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `z`
_warn_prf(average, modifier, msg_start, len(result))
```

Fig 58

In Fig 58, we can see the performance Metrics for the Decision Tree Model

6.2.4 Performance Metrics for KNN

```
[ ] print(classification_report(y_test,y_pred2))
```

	precision	recall	f1-score	support
0	0.50	1.00	0.67	1
1	0.00	0.00	0.00	1
accuracy			0.50	2
macro avg	0.25	0.50	0.33	2
weighted avg	0.25	0.50	0.33	2

```
/usr/local/lib/python3.9/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `z`
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.9/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `z`
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.9/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `z`
_warn_prf(average, modifier, msg_start, len(result))
```

Fig 59

In Fig 59, we can see the performance Metrics for the KNN Model.

6.2.5 Performance Metrics for MLP Classifier

```

} print(classification_report(y_test,y_pred))

              precision    recall  f1-score   support

     0       0.50      1.00      0.67         1
     1       0.00      0.00      0.00         1

 accuracy          0.25      0.50      0.33         2
 macro avg          0.25      0.50      0.33         2
 weighted avg          0.25      0.50      0.33         2

/usr/local/lib/python3.9/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `z
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.9/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `z
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.9/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `z
_warn_prf(average, modifier, msg_start, len(result))

```

Fig 60

In Fig 60, we can see the performance metrics for the MLP Classifier.

6. CONFUSION MATRIX FOR THE MODELS

6.3.1 Confusion Matrix for Random Forest

```

} from sklearn.metrics import confusion_matrix

} accuracy=confusion_matrix(y_test,predictions)

} accuracy

array([[1, 0],
       [1, 0]])

```

Fig 61

In Fig 61, we can see the Confusion Matrix for the Random Forest Classifier

6.3.2 Confusion Matrix for Gaussian Naïve Bayes

```

} accuracy1=confusion_matrix(y_test,y_pred)

} accuracy1

array([[1, 0],
       [1, 0]])

```

Fig 62

In Fig 62, we can see the confusion matrix for the Gaussian Naïve Bayes Model.

6.3.3 Confusion Matrix for Decision Tree

```

[ ] accuracy2=confusion_matrix(y_test,y_pred2)
accuracy2

array([[1, 0],
       [1, 0]])

```

Fig 63

In Fig 63, we can see the confusion matrix for Decision Tree

6.3.4 Confusion Matrix for KNN

```
] accuracy3=confusion_matrix(y_test,y_pred2)
accuracy3
array([[1, 0],
       [1, 0]])
```

Fig 64

In Fig 64, we can see the Confusion Matrix for KNN.

6.3.5 Confusion Matrix for MLP Classifier

```
[ ] accuracy5=confusion_matrix(y_test,y_pred3)
accuracy5

array([[1, 0],
       [1, 0]])
```

Fig 65

In Fig 65, we can see the Confusion Matrix for the MLP Classifier.

7. CONCLUSION

As a result of this project, we obtained the following results

Model	Accuracy
Random Forest	0.50
Gaussian Naïve Bayes	0.50
Decision Trees	0.50
K Nearest Neighbours	0.50
MLP Classifier	0.50

From these results, we can conclude that the detection of depression using Machine Learning is quite difficult as there are a lot of parameters to consider. In our project we only asked few specific questions and we overall got only around 90 responses which is what caused the accuracy to be so low for these models.

For the future of this project, we could ask more questions which are much more deep and also get more responses which would help the machine learn better.

In conclusion, using machine learning to detect depression has the potential to greatly improve mental health screening and treatment. Machine learning algorithms can analyze large amounts of data and identify patterns that might not be immediately visible to human clinicians, allowing for earlier diagnosis and more personalized treatment plans.

However, there are still several challenges to be addressed in the development and implementation of machine learning systems for depression detection. These include ensuring the privacy and security of patient data, addressing biases and inaccuracies in the algorithms, and ensuring that the technology is accessible and usable for both clinicians and patients.

Overall, machine learning has the potential to revolutionize the field of mental health, but it must be developed and implemented responsibly and ethically to ensure the best outcomes for patients.

9. REFERENCES:

1. Centers for Disease Control and Prevention, "National health and nutrition examination survey: Analytic guidelines, 2011–2016," pp. 1–40, 2018, [Online]. Available: <https://wwwn.cdc.gov/nchs/nhanes/analyticguidelines.aspx>
2. Y. Dong, E. C. Dragut, and W. Meng, "Normalization of duplicate records from multiple sources," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 4, pp. 769–782, Apr. 2019.
3. M. Masseroli, A. Canakoglu, and S. Ceri, "Integration and querying of genomic and proteomic semantic annotations for biomedical knowledge extraction," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 13, no. 2, pp. 209–219, Mar./Apr. 2016.
3. B. Ma, T. Jiang, X. Zhou, F. Zhao, and Y. Yang, "A novel data integration framework based on unified concept model," *IEEE Access*, vol. 5, pp. 5713–5722, 2017, doi: 10.1109/ACCESS.2017.2672822
5. T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biol. Cybern.*, vol. 43, no. 1, pp. 59–69, 1982,
6. New York, NY, USA: Wiley, 2001 V. Girotto and S. Pighin, "Basic understanding of posterior probability," *Front. Psychol.*, vol. 6, pp. 1–3, 2015, doi: 10.3389/fpsyg.2015.00680
7. M. Milic et al., "Tobacco smoking and health-related quality of life among university students: Mediating effect of depression", *PLoS One*, vol. 15, no. 1, pp. 1–18, 2020,
9. W. J. Zhang, C. Yan, D. Shum, and C. P. Deng, "Responses to academic stress mediate the association between sleep difficulties and depressive/anxiety symptoms in chinese adolescents," *J. Affect. Disorders*, vol. 263, pp. 89–98, Nov. 2019,
10. J. Davila, C. B. Stroud, L. R. Starr, M. R. Miller, A. Yoneda, and R. Hershenberg, "Romantic and sexual activities, parent-adolescent stress, and depressive symptoms among early adolescent girls," *J. Adolescence*, vol. 32, no. 4, pp. 909–924, 2009.

11. S. Y. Kim et al., “Gender and age differences in the association between work stress and incident depressive symptoms among Korean employees: A cohort study,” *Int. Archives Occupational Environmental Health*, vol. 93, no. 4, pp. 457–467, 2020, doi: 10.1007/s00420-019-01487-4.
12. M. Briley and J.-P. Lépine, “The increasing burden of depression,” *Neuropsychiatric Disease Treatment*, vol. 7, pp. 3–7, 2011,
13. A. H. Yazdavar et al., “Semi-supervised approach to monitoring clinical depressive symptoms in social media,” in *Proc. IEEE/ACM Int. Conf. Adv. Soc. Netw. Anal. Mining*, 2017, pp. 1191–1198
14. J. T. Wolohan, M. Hiraga, A. Mukherjee, and Z. A. Sayyed, “Detecting linguistic traces of depression in topic-restricted text: Attending to self-stigmatized depression with NLP,” *Workshop*, 2018, pp. 11–21,]
15. M. Srividya, S. Mohanavalli, and N. Bhalaji, “Behavioral modeling for mental health using machine learning algorithms,” *J. Med. Syst.*, vol. 42, no. 5, 2018, Art. no. 88, doi: 10.1007/s10916-018-0934-5