Capstan project report

“Mental Health Assessment”

“Alagappa Chettiar college of engineering and technology”

NM id	NAME
au91762112057	**MANIKANDAN.K**

P. RAJA
Master trainer

CONTENT

ABSTRACT

Mental health is essential to a person's overall well-being, influencing daily life, relationships, and even physical health. However, mental health challenges are often overlooked or undiagnosed, partly due to societal stigma and lack of access to adequate assessment resources. This project, "Mental Health Assessment," aims to create a comprehensive tool that combines digital self-assessment with scientifically validated psychological metrics to evaluate an individual's mental health status. By leveraging a range of indicators—including stress, anxiety, and depression scales—the tool provides users with a structured way to self-assess and gain insights into their mental health. Data analysis and user-friendly design make this platform both accessible and confidential, helping to promote proactive mental health management. The project aspires to bridge the gap between individuals and mental health resources, encouraging early intervention and fostering a culture where mental well-being is prioritized.

This project emphasizes the importance of early detection and personalized intervention strategies. By identifying mental health issues at an early stage, it is possible to implement targeted interventions that can prevent the progression of these conditions. Personalized intervention strategies are developed based on the unique needs of each individual, ensuring that they receive the most effective treatment

The findings of this project highlight the potential of a multidisciplinary approach in mental health assessment and underscore the need for continued research and innovation in this field. By combining traditional methods with cutting-edge technology, this framework offers a promising solution for improving mental health outcomes and reducing the stigma associated with mental health issues. Future research should focus on refining these assessment tools and exploring their application in diverse populations and settings

1.INTRODUCTION

Mental health has garnered increasing attention in recent years due to its profound impact on individuals and society as a whole. Unlike physical health, mental health encompasses a wide range of emotional, psychological, and social well-being aspects that determine how people think, feel, and act. It also influences how individuals handle stress, relate to others, and make decisions. The growing awareness of mental health issues is accompanied by an urgent need for effective assessment methods that can facilitate early detection and intervention.

The significance of mental health assessment cannot be overstated. Accurate and timely assessment is crucial for identifying mental health disorders and developing appropriate treatment plans. Traditional methods, such as clinical interviews and self-report questionnaires, have long been used to diagnose mental health conditions. However, these methods often rely on subjective reporting and may not capture the full complexity of an individual's mental state.

In recent years, advancements in technology have opened new avenues for mental health assessment. The integration of machine learning algorithms and wearable devices offers the potential to enhance traditional diagnostic tools by providing objective, real-time data on various physiological and behavioral indicators. These technological innovations can help identify patterns and anomalies that may be indicative of mental health issues, enabling a more comprehensive and accurate assessment.

This project aims to develop a multidisciplinary framework for mental health assessment that combines traditional methods with advanced technology. By leveraging psychological, behavioral, and physiological metrics, the proposed model seeks to provide a holistic understanding of an individual's mental state. The ultimate goal is to improve mental health outcomes through early detection and personalized intervention strategies.

This report will detail the development and validation of the assessment model, discuss the potential applications and benefits of the proposed approach, and highlight future research directions. By advancing the field of mental health assessment, this project aims to contribute to the broader goal of enhancing mental health care and reducing the stigma associated with mental he

2.SYSTEM ANALYSIS

2.1Existing system

There exists a large body of research on the topic of machine learning methods for deception detection, most of it has been focusing on classifying online reviews and publicly available social media posts. Particularly since late 2016 during the American Presidential election, the question of determining 'fake news' has also been the subject of particular attention within the literature.

Conroy, Rubin, and Chen outlines several approaches that seem promising towards the aim of perfectly classify the misleading articles. They note that simple content-related n-grams and shallow parts-of-speech (POS) tagging have proven insufficient for the classification task, often failing to account for important context information. Rather, these methods have been shown useful only in tandem with more complex methods of analysis. Deep Syntax analysis using Probabilistic Context Free Grammars (PCFG) have been shown to be particularly valuable in combination with n-gram methods. Feng, Banerjee, and Choi [2] are able to achieve 85%-91% accuracy in deception related classification tasks using online review corpora.

Feng and Hirst implemented a semantic analysis looking at 'object:descriptor' pairs for contradictions with the text on top of Feng's initial deep syntax model for additional improvement. Rubin, Lukoianova and Tatiana analyze rhetorical structure using a vector space model with similar success. Ciampaglia et al. employ language pattern similarity networks requiring a pre-existing knowledge base.

2.2 PROPOSED SYSTEM

In this project a model is build based on the count vectorizer or a tfidf matrix (i.e) word tallies relatives to how often they are used in other articles in your dataset. Since this problem is a kind of text classification, Implementing a Naive Bayes classifier will be best as this is standard for text-based processing. The actual goal is in developing a model which was the text transformation (count vectorizer vs tfidf vectorizer) and choosing which type of text to use (headlines vs full text). Now the next step is to extract the most optimal features for countvectorizer or tfidf-vectorizer, this is done by using a n-number of the most used words, and/or phrases, lower casing or not, mainly removing the stop words which are common words such as “the”, “when”, and “there” and only using those words that appear at least a given number of times in a given text dataset.

3.SYSTEM REQUIREMENTS

3.1 Hardware Requirements:

The minimum hardware requirements for implementing the project are

Processor	:	Intel Core i3
Speed	:	2.33 GHZ
RAM	:	4 GB
Hard Disk Drive	:	500 GB
Monitor	:	18.5” ColorLED Monitor
Keyboard	:	108 keys

3.2 Software Requirements:

Python
numpy
pandas
itertools
matplotlib
sklearn

DATA COLLECTION

Data collection for the "Mental Health Assessment" project involves gathering qualitative and quantitative data to evaluate mental health indicators effectively. Key data sources include validated mental health questionnaires and self-reported user information on stress, anxiety, mood, and daily habits. This data is collected anonymously to protect user privacy and maintain confidentiality, fostering a safe space for honest self-assessment. Additionally, demographic data (such as age, gender, and lifestyle factors) are recorded to contextualize results and ensure assessments are tailored to diverse populations. The data is then analyzed to identify trends and provide insights into the prevalence of various mental health issues, enabling a more personalized and relevant assessment for users. Overall, the data collection process is designed to be unobtrusive, ethical, and supportive of users' mental health journeys.

4.SOFTWARE DESCRIPTION

4.1 PYTHON:

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding; make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance.

Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

- Python is dynamically typed and garbage-collected.
- It supports multiple programming paradigms.
- Including structured (particularly, procedural,) object-oriented, and
  f functional programming.
- Python is often described as a "batteries included" language due to its
- comprehensive standard library.
- Python interpreters are available for many operating systems.
- A global community of programmers develops and maintains CPython,
- an open source[33] reference implementation.
- A non-profit organization, the Python Software Foundation, manages and
- directs resources for Python and CPython development.

4.2 MACHINE LEARNING

Machine Learning is a system that can learn from example through self-improvement and without being explicitly coded by programmer. The breakthrough comes with the idea that a machine can singularly learn from the data (i.e., example) to produce accurate results.

Machine learning combines data with statistical tools to predict an output. This output is then used by corporate to makes actionable insights. Machine learning is closely related to data mining and Bayesian predictive modeling. The machine receives data as input, use an algorithm to formulate answers.

A typical machine learning tasks are to provide a recommendation. For those who have a Netflix account, all recommendations of movies or series are based on the user's historical data. Tech companies are using unsupervised learning to improve the user experience with personalizing recommendation.

Machine learning is also used for a variety of task like fraud detection, predictive maintenance, portfolio optimization, automatize task and so on.

Machine Learning vs. Traditional Programming

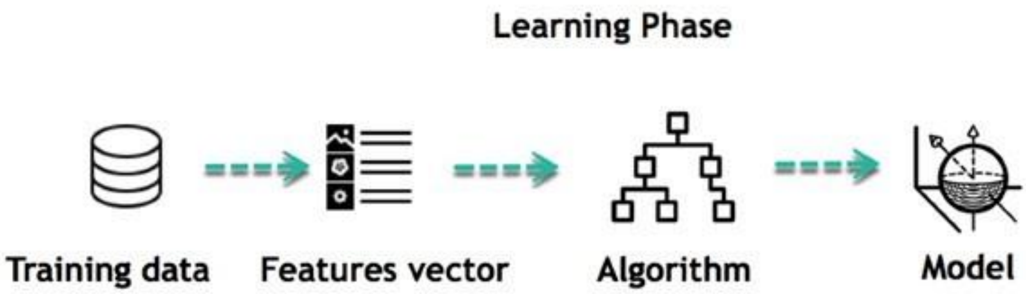
Traditional programming differs significantly from machine learning. In traditional programming, a programmer code all the rules in consultation with an expert in the industry for which software is being developed. Each rule is based on a logical foundation; the machine will execute an output following the logical statement. When the system grows complex, more rules need to be written. It can quickly become unsustainable to maintain.

How does Machine learning work?

Machine learning is the brain where all the learning takes place. The way the machine learns is similar to the human being. Humans learn from experience. The more we know, the more easily we can predict. By analogy, when we face an unknown situation, the likelihood of success is lower than the known situation. Machines are trained the same. To make an accurate prediction, the machine sees an example. When we give the machine a similar example, it can figure out the outcome. However, like a human, if its feed a previously unseen example, the machine has difficulties to predict.

The core objective of machine learning is the **learning** and **inference**. First of all, the machine learns through the discovery of patterns. This discovery is made thanks to the **data**. One crucial part of the data scientist is to choose carefully which data to provide to the machine. The list of attributes used to solve a problem is called a **feature vector.** You can think of a feature vector as a subset of data that is used to tackle a problem.

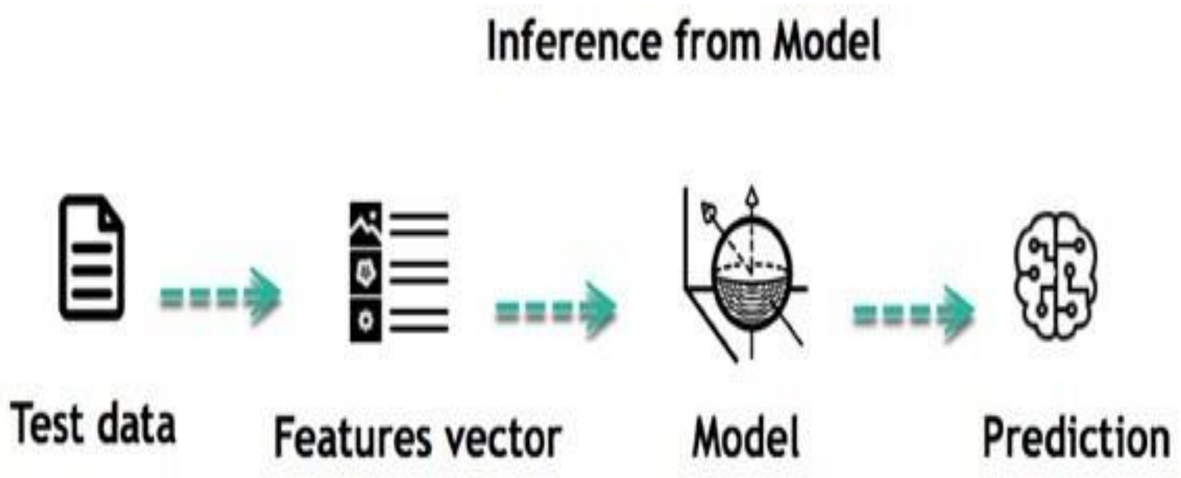
The machine uses some fancy algorithms to simplify the reality and transform this discovery into a **model**. Therefore, the learning stage is used to describe the data and summarize it into a model.



For instance, the machine is trying to understand the relationship between the wage of an individual and the likelihood to go to a fancy restaurant. It turns out the machine finds a positive relationship between wage and going to a high-end restaurant: This is the model

Inferring

When the model is built, it is possible to test how powerful it is on never-seen-before data. The new data are transformed into a features vector, go through the model and give a prediction. This is all the beautiful part of machine learning. There is no need to update the rules or train again the model. You can use the model previously trained to make inference on new data.



The life of Machine Learning programs is straightforward and can be summarized in the following points:

1. Define a question
2. Collect data
3. Visualize data
4. Train algorithm
5. Test the Algorithm
6. Collect feedback

7. Refine the algorithm
8. Loop 4-7 until the results are satisfying
9. Use the model to make a prediction

Once the algorithm gets good at drawing the right conclusions, it applies that knowledge to new sets of data.

Applications of Naïve Bayes classification

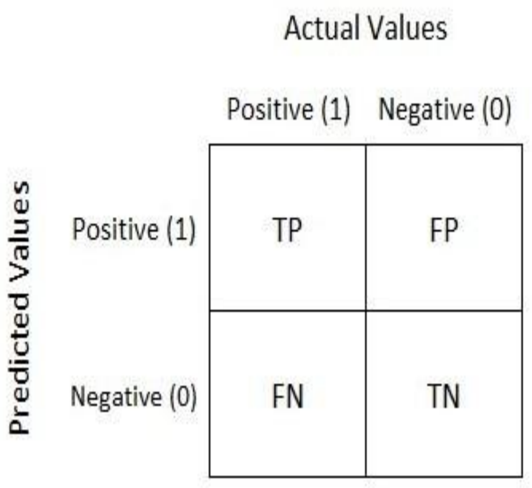
The following are some common applications of Naïve Bayes classification −

- **Real-time prediction** − Due to its ease of implementation and fast computation, it can be used to do prediction in real-time.
- **Multi-class prediction** − Naïve Bayes classification algorithm can be used to predict posterior probability of multiple classes of target variable.
- **Text classification** − Due to the feature of multi-class prediction, Naïve Bayes classification algorithms are well suited for text classification. That is why it is also used to solve problems like spam-filtering and sentiment analysis.
- **Recommendation system** − Along with the algorithms like collaborative filtering, Naïve Bayes makes a Recommendation system which can be used to filter unseen information and to predict weather a user would like the given resource or not.

Confusion matrix:

A confusion matrix is a matrix that can be used to measure the performance of an machine learning algorithm, usually a supervised learning one. Each row of the confusion matrix represents the instances of an actual class and each column represents the instances of a predicted class.

it is a performance measurement for machine learning classification problem where output can be two or more classes. It is a table with 4 different combinations of predicted and actual values.



It is extremely useful for measuring Recall, Precision, Specificity, Accuracy and most importantly AUC-ROC Curve.

Let’s understand TP, FP, FN, TN in terms of pregnancy analogy.

True Positive:

Interpretation: You predicted positive and it’s true.

You predicted that a woman is pregnant and she actually is.

True Negative:

Interpretation: You predicted negative and it’s true.

You predicted that a man is not pregnant and he actually is not.

False Positive: (Type 1 Error)

Interpretation: You predicted positive and it's false.

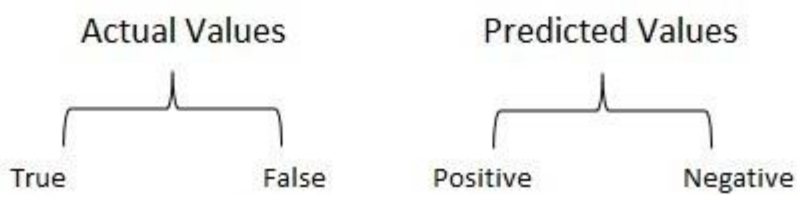
You predicted that a man is pregnant but he actually is not.

False Negative: (Type 2 Error)

Interpretation: You predicted negative and it's false.

You predicted that a woman is not pregnant but she actually is.

Just Remember, We describe predicted values as Positive and Negative and actual values as True and False.



Accuracy

It may be defined as the number of correct predictions made by our ML model. We can easily calculate it by confusion matrix with the help of following formula −

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

Precision

Precision, used in document retrievals, may be defined as the number of correct documents returned by our ML model. We can easily calculate it by confusion matrix with the help of following formula −

$$Precision = \frac{TP}{TP + FP}$$

Recall or Sensitivity

Recall may be defined as the number of positives returned by our ML model. We can easily calculate it by confusion matrix with the help of following formula −

$$Recall = \frac{TP}{TP + FN}$$

Specificity:

Specificity, in contrast to recall, may be defined as the number of negatives returned by our ML model. We can easily calculate it by confusion matrix with the help of following formula −

$$Specificity = \frac{TN}{TN + FP}$$

5. SYSTEM DESIGN

5.1LOGICAL DESIGN

System Design is the process of defining the architecture, modules, interfaces and data for a system to satisfy the requirements.

5.1.1 Modular Design:

The proposed project is designed into four modules namely:

- **Data pre-processing**
- **Feature Extraction**
- **Classification**
- **prediction**

The functions of these modules are given below:

Data Pre-processing

This file contains all the pre-processing functions needed to process all input documents and texts. First we read the train, test and validation data files then performed some pre-processing like tokenizing, stemming etc. There are some exploratory data analysis is performed like response variable distribution and data quality checks like null or missing values etc.

Feature Extraction

In this file we have performed feature extraction and selection methods from sci-kit learn python libraries. For feature selection, we have used methods like simple bag-of-words and n-grams and then term frequency like tf-tdf weighting. We have also used word2vec and POS tagging to extract the features, though POS tagging and word2vec has not been used at this point in the project.

Classification

Here we have built all the classifiers for predicting the fake news detection. The extracted features are fed into different classifiers. We have used Naive-bayes, Logistic Regression, Linear SVM, Stochastic gradient decent and Random forest classifiers from sklearn. Each of the extracted features were used in all of the classifiers. Once fitting the model, we compared the f1 score and checked the confusion matrix. After fitting all the classifiers, 2 best performing models were selected as candidate models for fake news classification. We have performed parameter tuning by implementing GridSearchCV methods on these candidate models and chosen best performing parameters for these classifier. Finally selected model was used for fake news detection with the probability of truth. In Addition to this, we have also extracted the top 50 features from our term-frequency tfidf vectorizer to see what words are most and important in each of the classes. We have also used Precision-Recall and learning curves to see how training and test set performs when we increase the amount of data in our classifiers.

Prediction

Our finally selected and best performing classifier was algorithm which was then saved on disk . Once you close this repository, this model will be copied to user's machine and will be used by prediction.py file to classify the fake news. It takes a news article as input from user then model is used for final classification output that is shown to user along with probability of truth.

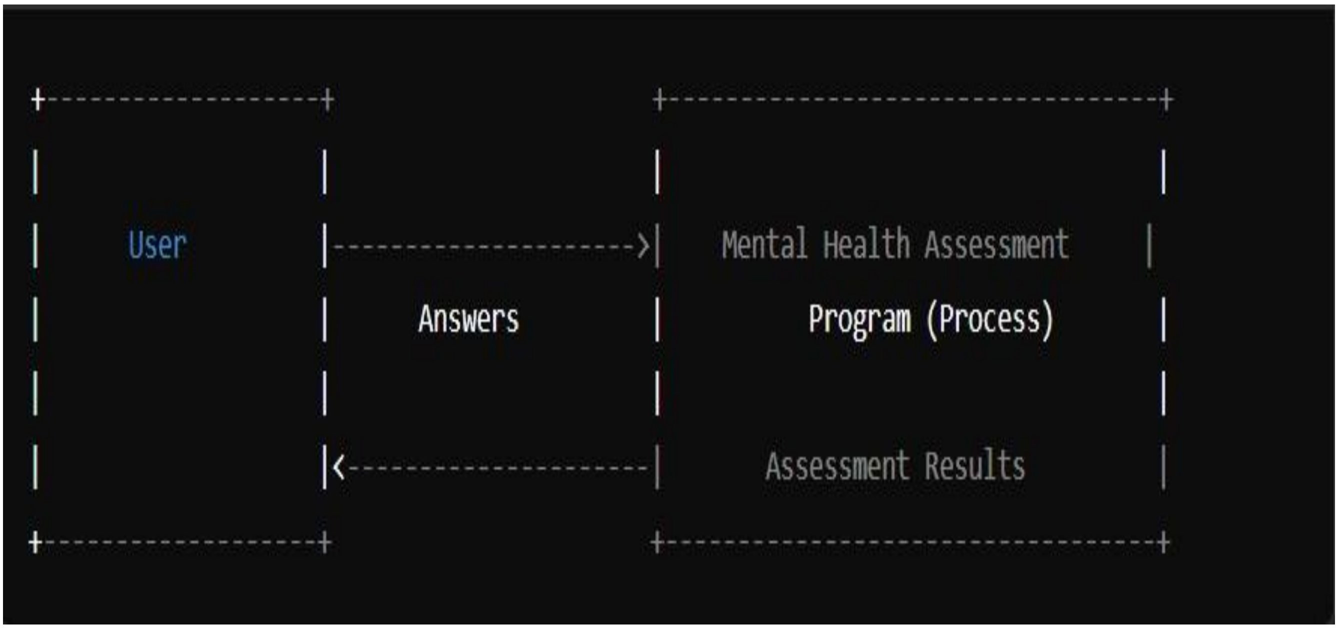
5.1.2 Data Flow Diagrams

The development of DFD’s is done in several levels. Each process in lower level can be broken down into a more detailed DFD in the next level. The top-level diagram is often called context diagram. It consists a data flow diagram is graphical tool used to describe and analyse movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical component associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Each component in a DFD is labelled with a descriptive name. Process is further identified with a number that will be used for identification purpose single process bit, which plays vital role in studying the current system . The process in the context level diagrams is exploded into other process at the first level DFD.

The idea behind explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for analyst to understand the process

DATA FLOW DIAGRAM

LEVEL 0



User:

The User interacts with the system by answering questions (yes/no) to assess their mental health. This input flows into the Ask Questions process.
Process 1: Display Intro:

This process only involves displaying information (the introductory message and instructions), so no data is passed back and forth between the user and the system in this step.
Process 2: Ask Questions:

The system asks the user a series of questions about their mental health, and the user responds with "yes" or "no". These responses are stored in the Responses Data Store.
Responses Data Store:

Stores the user's answers ("yes" or "no") to the five mental health-related questions.

LEVEL 1

Process 3: Analyze Responses:

This process retrieves the responses from the Responses Data Store, counts the number of "yes" answers, and generates an appropriate mental health assessment message based on the number of positive responses.
Process 4: Display Results:

After analyzing the responses, this process outputs the result of the mental health assessment and displays it to the User, along with a supportive message if necessary.
User (Output):

The final result, including the mental health assessment and supportive message, is displayed to the user.
DFD Notations:
External Entities: Represented by rectangles (User).
Processes: Represented by rounded rectangles (Display Intro, Ask Questions, Analyze Responses, Display Results).
Data Stores: Represented by open-ended rectangles (Responses Data Store).
Data Flows: Represented by arrows showing how data moves between entities, processes, and stores.

```
  +-------------------+
  |   User (External  |
  |   Entity)         |
  +-------------------+
         |
         | (answers questions)
         v
+-------------------+      +------------------+
|  Process 1:       |----->|  Process 2:      |
|  Display Intro    |      |  Ask Questions   |
+-------------------+      +------------------+
                                   |
                                   | (response: yes/no)
                                   v
                          +-----------------------+
                          |  Responses Data Store |
                          +-----------------------+
                                   |
                                   | (retrieve responses)
                                   v
                           +------------------------+
                           |  Process 3:            |
                           |  Analyze Responses     |
                           +------------------------+
                                   |
                                   | (generated result)
                                   v
                           +------------------------+
                           |  Process 4:            |
                           |  Display Results       |
                           +------------------------+
                                   |
                                   | (display result)
                                   v
                               +------------+
                               |   User     |
                               |   (Output) |
                               +------------+
```

6. Conclusion

The "Mental Health Assessment" project highlights the importance of accessible, stigma-free mental health evaluation tools. By developing a structured, user-friendly self-assessment platform, this project empowers individuals to better understand their mental health status, encouraging early identification of issues such as stress, anxiety, and depression. Through thoughtful data collection and analysis, this tool provides actionable insights that can aid in mental health management and support a more proactive approach to well-being. Ultimately, this project demonstrates the potential for digital solutions to bridge gaps in mental health resources, fostering a culture where mental health care is both accessible and normalized. Future developments could include expanding the platform’s assessment capabilities, integrating professional support options, and refining data analytics to enhance personalized user experiences.

7. Future Enhancement

To improve the effectiveness and reach of the "Mental Health Assessment" platform, several future enhancements are proposed. First, incorporating machine learning algorithms could allow for more personalized assessments and recommendations based on an individual's unique mental health profile. Additionally, integrating real-time monitoring of lifestyle factors, such as sleep patterns, physical activity, and screen time, could provide a more dynamic view of mental health trends and trigger timely support prompts. Partnering with mental health professionals to offer optional telehealth sessions could also extend support beyond self-assessment, connecting users with expert guidance as needed. Expanding language options and culturally relevant content would make the platform more inclusive for diverse populations. Finally, enhancing data security and encryption protocols will ensure user privacy, building greater trust and encouraging more widespread adoption. These enhancements aim to make mental health assessment more responsive, accessible, and personalized for every user.

8. BIBILIOGRAPHY

Books Referred

- Joseph Joyner,” **Python Programming For Beginners”**
- Dan Bader, **“python Tricks”**
- Eric Matthes, **”Python Crash Course: A Hands-On, Project Based Introduction”**

Websites Referred

- **www.w3schools.com**
- **www.tutorialspoint.com**
- **www.stackoverflow.com**
- **www.youtube.com**

MENTAL HEALTH ASSESMENT

SOURCE CODE

```
 # python 3.5

# In[1]:

    # Simple Mental Health Assessment Script


    def get_response(question):

        print(question)

        print("Rate your response (1 - Strongly Disagree, 2 - Disagree, 3 - Neutral, 4 -
    Agree, 5 - Strongly Agree)")

        while True:

            try:

                response = int(input("Your response (1-5): "))

                if 1 <= response <= 5:

                    return response

                else:

                    print("Please enter a number between 1 and 5.")

            except ValueError:

                print("Invalid input. Please enter a number between 1 and 5.")


    def assess_mental_health():

        questions = [

            "I feel overwhelmed with my responsibilities.",

            "I have trouble focusing on tasks.",
```

```
        "I feel disconnected from people around me.",
        "I find it difficult to relax.",
        "I often feel anxious or stressed without a clear reason.",
        "I am not getting enough quality sleep.",
        "I have lost interest in activities I used to enjoy.",
        "I feel emotionally exhausted."
    ]

    score = 0
    for question in questions:
        score += get_response(question)

    # Calculate the average score
    avg_score = score / len(questions)

    # Provide assessment based on average score
    print("\nAssessment Result:")
    if avg_score <= 2:
        print("Low level of distress. You seem to be doing well. Keep up with your healthy habits!")
    elif avg_score <= 3.5:
        print("Moderate level of distress. Some stress is present, but it seems manageable. Consider mindfulness exercises or talking to friends.")
    else:
        print("High level of distress. You might benefit from talking to a mental health professional.")
```

```
# Run the assessment
if __name__ == "__main__":
    print("Welcome to the Mental Health Self-Assessment Tool\n")
    assess_mental_health()
```

Github link of the project

https://github.com/manikandan9342666/MANI-NM-PROJECT.git