AUTOMATED STRESS LEVEL ASSESSMENT BY EXPLORING PREDICTIVE TECHNIQUES

A PROJECT REPORT

Submitted by

ISHWARYA R (2127200501059)
JESSICA JANET P A (2127200501063)
PREETI PURNIMAA K (2127200501111)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

SRI VENKATESWARA COLLEGE OF ENGINEERING
(An Autonomous Institution; Affiliated to Anna University, Chennai-600025)
ANNA UNIVERSITY :: CHENNAI 600 025

MAY 2024

SRI VENKATESWARA COLLEGE OF ENGINEERING
(An Autonomous Institution; Affiliated to Anna University, Chennai-600025)

ANNA UNIVERSITY :: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report "**AUTOMATED STRESS LEVEL ASSESSMENT BY EXPLORING PREDICTIVE TECHNIQUES"** is the bonafide work of **"ISHWARYA R (2127200501059), JESSICA JANET P A (2127200501063)** and **PREETI PURNIMAA K (2127200501111)**" who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr.R.ANITHA	**Dr. P. JANARTHANAN**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
	PROFESSOR
COMPUTER SCIENCE & ENGG	**COMPUTER SCIENCE & ENGG**

Submitted for the project viva-voce examination held on ………………..

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

Stress is a prevalent issue affecting individuals across various domains of life, including work, education, and personal relationships. The objective of this research is to develop a reliable and accurate system that can predict stress levels based on various input features. The study employs supervised learning techniques, which utilize labelled training data to build a predictive model. The model is trained to classify stress levels based on the input features and corresponding stress labels. The proposed methodology involves several stages. Firstly, relevant features related to stress are identified, such as heart rate, sleep patterns, physical activity, and self-reported emotions. These features are extracted from individuals using wearable devices, surveys, and other data collection methods. The implications of this research are significant, as an accurate stress level analysis system can aid in identifying individuals at risk of high stress and enable timely interventions. It can be utilized in various settings, including healthcare, workplaces, and educational institutions, to promote well-being and prevent stress-related health issues. In conclusion, this study aims to develop a stress level analysis system using supervised learning approaches. The results of this research can contribute to the development of personalized stress management interventions, leading to improved overall mental health and well-being for individuals.

ACKNOWLEDGEMENT

We thank our Principal **Dr. S. Ganesh Vaidyanathan,** Sri Venkateswara College of Engineering for being the source of inspiration throughout our study in this college.

We express our sincere thanks to **Dr. R. Anitha,** Professor and Head of the Department, Computer Science and Engineering for her encouragement accorded to carry this project.

With profound respect, we express our deep sense of gratitude and sincere thanks to our guide **Dr. P. Janarthanan**, **Professor**, for his valuable guidance and suggestions throughout this project.

We are also thankful to our Project coordinators **Dr. R. Jayabhaduri, Professor, Dr.N.M.Balamurugan, Professor and Dr.V.Rajalakshmi, Associate Professor** for their continual support and assistance.

We thank our family and friends for their support and encouragement throughout the course of our graduate studies.

ISHWARYA R
JESSICA JANET P A
PREETI PURNIMAA K

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

API	Application Programming Interface
GUI	Graphical User Interface
HTML	Hyper Text Markup Language
IDE	Integrated Development Environment
IOT	Internet of Things
JSON	JavaScript Object Notation
LSVM	Linear Support Vector Machine
MSE	Mean Square Error
RAM	Random Access Memory
SGD	Stochastic Gradient Descent
YAML	Yet Another Markup Language

CHAPTER 1

INTRODUCTION

1.1 PROJECT DEFINITION

This project aims at utilizing predictive techniques in machine learning to determine stress levels. The main focus is on developing a model that can accurately predict stress levels based on various input factors. By leveraging advanced machine learning algorithms, we aim to provide a reliable and efficient tool for assessing and managing stress in individuals.

The proposed work aims contribute to the field of mental health and well-being by offering a data-driven approach to understanding and addressing stress. The application is built using several technologies, a modern, fast web framework for building APIs creating interactive data science applications.

1.2 NEED FOR PROPOSED WORK

Stress level analysis using supervised learning is a proposed system that aims to predict and analyse an individual's stress levels based on various input features. Supervised learning refers to a machine learning approach where a model is trained on labeled data to make predictions or classifications. The predicted stress levels can be further analyzed to gain insights into patterns, triggers, or factors contributing to stress to achieve stress analysis and insights.

Data visualization techniques and statistical analyses can help identify trends or correlations between features and stress levels. Data is collected and analyzed, and the system can provide personalized feedback and recommendations to individuals based on their stress levels. This feedback can include stress management strategies, relaxation techniques, or lifestyle adjustments to help reduce stress and improve well-being.

1.3 APPLICATION OF PROPOSED SYSTEM

The scope of human stress analysis encompasses a wide range of factors related to the physiological, psychological, and environmental aspects of stress experienced by individuals which results to numerous applications in various sectors. In this section, we will discuss the potential applications of the system in different sectors.

1. Pysiological Responses: In addition to studying changes in heart rate, blood pressure, and other physiological markers, researchers also investigate the impact of stress on the autonomic nervous system, hormonal responses (such as cortisol levels), and inflammatory markers. Understanding these physiological responses can provide valuable insights into the body's reaction to stress and potential health implications.

2. Psychological Responses: Research in this area delves into the cognitive and emotional responses to stress, including the role of coping mechanisms, resilience factors, and the development of stress-related disorders. Studies may explore the effects of chronic stress on cognitive function, memory, decision-making, and overall mental well-being.

3. Risk Factors and Protective Factors: Identifying specific risk factors that contribute to increased vulnerability to stress, such as genetic predispositions, early life experiences, and environmental stressors, is crucial for developing targeted interventions. Conversely, research on protective factors, such as social support, coping strategies, and mindfulness practices, can help mitigate the negative impact of stress on individuals.

4. Health Consequences: Chronic stress has been linked to a myriad of adverse health outcomes beyond cardiovascular diseases, immune dysfunction, gastrointestinal issues, and mental health disorders. Additional health consequences may include sleep disturbances, metabolic disorders, chronic pain conditions, and accelerated aging processes. Understanding the full spectrum of health consequences associated with stress is essential for comprehensive intervention strategies.

5. Assessment Tools and Measurement: Developing reliable and valid measures for assessing stress is a fundamental aspect of both research and clinical practice. Researchers may utilize self-report questionnaires, physiological monitoring devices, neuroimaging techniques, and biomarker assessments to capture the multidimensional nature of stress. Validated assessment tools enable accurate diagnosis, treatment planning, and monitoring of stress-related conditions, ultimately enhancing the quality of care provided to individuals experiencing stress.

CHAPTER 2

LITERATURE REVIEW

Jinwoo Park, Hongryul Ahn , Kanwoo Youn, Minji Lee, And Seunghyeok Hong (2023) addressed the study conducted on understanding the factors contributing to depression among farmers is of paramount importance for ensuring their well-being and productivity. By employing advanced tree-based machine learning algorithms, specifically focusing on the Category Boosting (CatB) algorithm, the research has provided valuable insights into this critical issue. Through the application of the Patient Health Questionnaire-9 (PHQ-9) criteria, a total of 2,446 individuals out of 14,810 respondents were classified into depression, including mild symptoms. The CatB algorithm demonstrated an impressive 79.7% accuracy and 81.4% F1 score in the classification task, outperforming other tree-based ensemble models such as Random Forest (RF), Extra Trees (ET), and XGBoost (XGB). Notably, RF exhibited the highest sensitivity at 90.0% and an 81.3% F1 score, closely followed by CatB. In terms of feature importance analysis, the Gini impurity metric was predominantly utilized to assess the Random Forest and Extra Trees models. The analysis revealed that factors such as 'Health', 'Sleep time', 'Busyness', 'Income', and 'Frequency of wearing protective gear' were identified as significant features contributing to depression among farmers. These findings underscore the importance of developing targeted treatment strategies for individuals at high risk of depression within the agricultural sector. By empowering healthcare providers with access to this tool and the insights derived from the study, more effective interventions can be implemented, potentially reducing the burden of de pression and enhancing the overall productivity and well-being of farmers. This research serves as a valuable resource for addressing mental health challenges in the farming community and promoting a healthier and more resilient agricultural workforce.

Dennis R. da C. Silva , Zelun Wang , and Ricardo Gutierrez-Osuna (2021) focused on introducing a minimalist design that offers simplicity and cost-effectiveness, making it accessible for replication by other researchers. By utilizing off-the-shelf and low-cost hardware, the proposed design ensures that the stress-detection system can be easily implemented in various settings. The validation of this design is conducted through a laboratory experiment that replicates office tasks and mild stressors, providing a controlled environment for testing. This approach allows researchers to avoid the methodological limitations that have been observed in previous studies, ensuring the reliability and validity of the results. In the experiment, the stress-detection performance of the minimalist design is compared with existing conventional features that have been previously reported in the literature. By conducting this comparison, the study aims to evaluate the effectiveness and accuracy of the proposed design in detecting stress levels in individuals. This rigorous evaluation process contributes to the advancement of stress-detection technology and provides valuable insights for future research and development in this field.

Karl Magtibay,and Karthikeyan Umapathy, Senior Member, IEEE (2023) focused on the study underscores the significance of integrating the allostatic load model into affective computing for the development of disease and illness prediction models. To elaborate further, let's delve into the physiological and psychological foundations of allostasis and its relevance to stress studies within affective computing: Allostatic load, a key component of the allostatic model, represents the cumulative wear and tear on the body resulting from prolonged or excessive stress responses. By considering multiple physiological parameters, such as cortisol levels, heart rate variability, and inflammatory markers, the allostatic load model offers a comprehensive framework for assessing the impact of chronic stress on health outcomes. The allostatic load model can enhance the accuracy and predictive power of stress-related studies. By incorporating physiological markers of stress into computational models, researchers can gain deeper insights into the complex interplay between emotional states, physiological responses, and health outcomes. By leveraging the allostatic load model in affective computing research, scholars can develop These models have the potential to not only identify individuals at risk for stress-related disorders but also inform targeted interventions and preventive strategies to mitigate the negative impact of chronic stress on health and well-being. In summary, integrating the allostatic load model into affective computing represents a promising avenue for advancing our understanding of stress, health, and disease prediction. By bridging the gap between physiological and psychological factors, researchers can pave the way for more effective and personalized approaches to managing stress and promoting overall health.

Hena Yasmin, Swaziland, Salman Khalil, Amity University, India, Ramsha Mazhar, Allan Grey, SouthAfrica (2020) focused on the global COVID-19 pandemic has undoubtedly brought about unprecedented levels of distress and anxiety across all age groups, including children and students.The disruption caused by the pandemic, such as sudden transitions to remote learning, social isolation, uncertainty about the future, and health concerns, has significantly contributed to heightened stress levels among students. These stressors can manifest in various ways, including decreased academic performance, emotional instability, behavioral issues, and overall well-being. Understanding the impact of stress on students is crucial in order to address these challenges effectively. By exploring the relationship between stress and learning outcomes, we can identify strategies and interventions to support students in managing their stress levels and promoting a conducive learning environment. Effective stress management techniques, such as mindfulness practices, relaxation exercises, counseling services, and social support systems, can play a vital role in enhancing students' resilience and well-being. By emphasizing the necessity of managing stress among students, we aim to underscore the importance of prioritizing mental health and well-being in educational settings. Through proactive measures and targeted interventions, we can create a supportive and nurturing environment that enables students to thrive academically, emotionally, and socially. This paper serves as a call to action to recognize and address the impact of stress on students and to implement holistic approaches to promote effective learning and overall student success.

Mr. Devraj Singh Chouhan (2016) addressed that in recent years, the medical sector has made significant advancements in various areas, yet there remains a noticeable gap in addressing mental health issues. Surveys conducted in recent years have highlighted the prevalence of psychological problems, which significantly impact the quality of life and physical health of individuals. Stress, in particular, is a common response to stimuli that can have detrimental effects on the body's mechanisms and disrupt its nutritional patterns. This paper aims to contribute to improving knowledge and awareness surrounding mental health and effective stress management strategies. By delving into the complexities of mental health and stress, it seeks to shed light on the importance of addressing these issues proactively. Understanding the impact of stress on the body and mind is crucial in developing holistic approaches to maintaining overall well-being. Through this paper it aims to provide insights into coping mechanisms, resilience-building strategies, and the importance of seeking professional help when needed. By raising awareness and offering practical guidance, we hope to empower individuals to take control of their mental health and lead healthier, more fulfilling lives. Together, we can work towards destigmatizing mental health issues and promoting a culture of self-care and support.

Amir Mohammad Shahsavarani, Esfandiar Azad marz Abadi, Maryam Hakimi Kalkhoran (2016) noted that the design of this study involved a systematic review methodology, which aimed to comprehensively analyze existing literature on the subject of stress and related topics. Inclusion criteria were established to ensure that only studies relevant to the research objectives were considered. Specifically, studies were included if they addressed key keywords related to stress, such as "stress," "stress control," "stress reduction," "social stress," "community stress," and "group stress. "By focusing on these specific keywords, the systematic review aimed to capture a wide range of studies that explored different aspects of stress management and its impact on individuals and communities. This approach allowed for a thorough examination of the existing body of literature on stress, providing valuable insights into the various strategies and interventions used to address stress in different contexts. Through the systematic review process, the study aimed to synthesize and analyze the findings from the selected studies to identify common themes, trends, and gaps in the literature. By expanding the search criteria to include a diverse range of stress-related keywords, the study sought to provide a comprehensive overview of the current knowledge and research landscape in the field of stress management.

CHAPTER 3

PROPOSED WORK

3.1 MAIN OBJECTIVE

The main objective of this project is to develop a machine learning-based stress prediction system that can provide stress management plans to individuals based on their level of stress. The system aims to improve the accuracy and effectiveness of stress management strategies, which can lead to better mental health outcomes and reduced risk of stress-related disorders. There are several reasons why a personalized stress prediction system is needed.

Firstly, there is a growing recognition of the impact of stress on mental health and overall well-being. Predicting stress levels can help individuals proactively manage their stress and prevent the onset of mental health issues.

Secondly, individuals have different stress triggers and coping mechanisms, making a one-size-fits-all approach to stress management ineffective. A personalized approach that considers individual stressors and coping strategies can enhance the effectiveness of stress management interventions.

Thirdly, traditional methods of stress assessment and counseling may be limited in scalability and accessibility. A machine learning-based stress prediction system has the potential to provide personalized stress management advice on a large scale, making it more accessible and cost-effective.

Therefore, the main objective of the stress prediction system is to develop a technology-based solution that can accurately predict stress levels and provide personalized stress management plans to individuals, ultimately contributing to better mental health outcomes.

3.2 PROPOSED ARCHITECTURE

Figure 3.2 shows the architecture for the proposed model for predicting the stress levels.

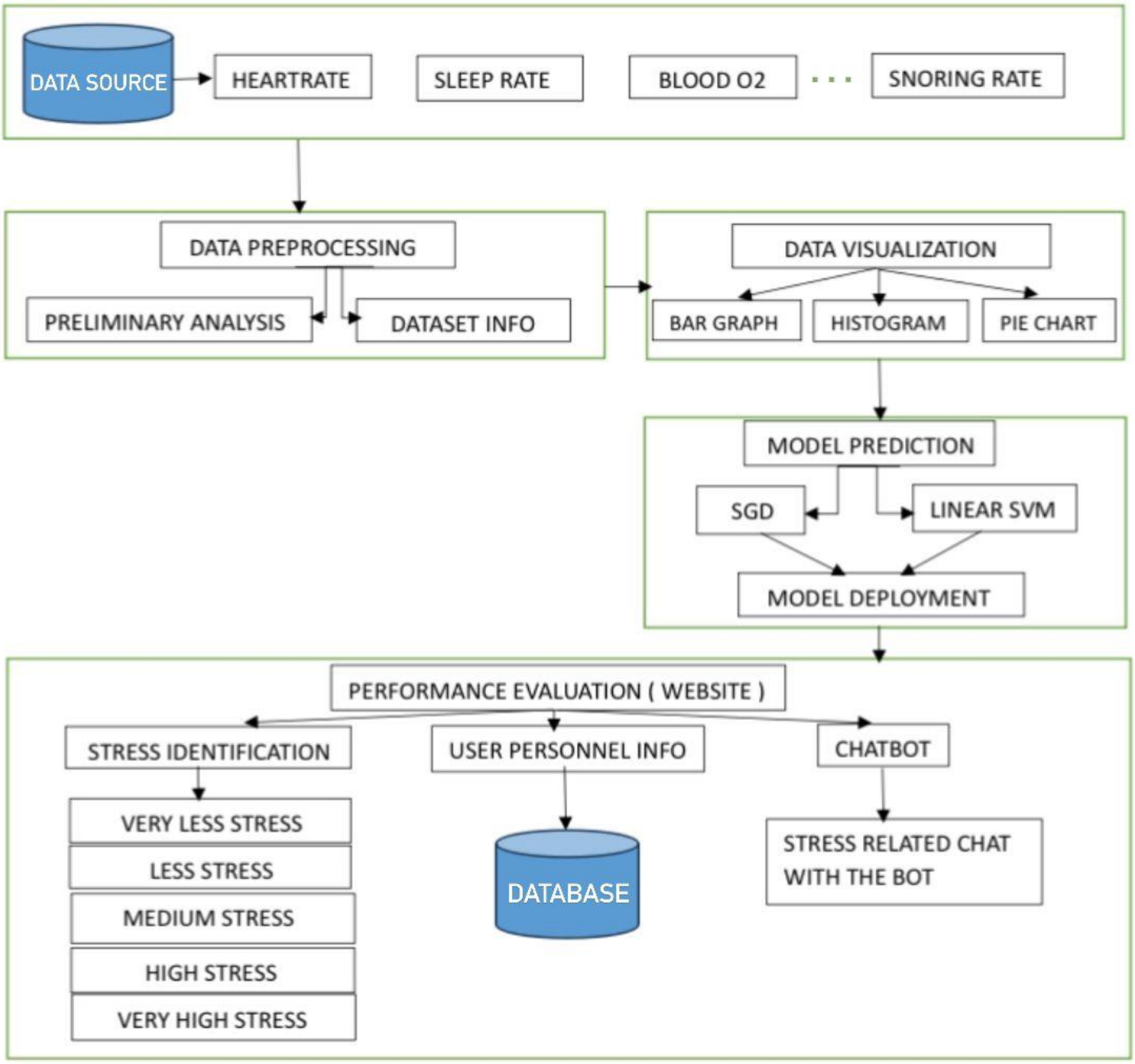


Figure 3.2 Proposed Architecture

3.3 COMPONENTS OF THE ARCHITECTURE

3.3.1 Data Source

The Dataset for the project was collected from Kaggle.com. This component represents a database that stores stress prediction components. It includes details such as snoring rate, respiration rate, temperature, limb movement, blood oxygen, eye movement, sleeping hours, heart rate, stress level. The Database acts as a data source for predicting stress level.

3.3.2 Data Preprocessing

A number of different data cleaning tasks using Python's Pandas library and specifically, it focus on probably the biggest data cleaning task, missing values and it able to more quickly clean data. It wants to spend less time cleaning data, and more time exploring and modeling. Some of these sources are just simple random mistakes. The type of missing data will influence how to deal with filling in the missing values and to detect missing values, and do some basic implementation and detailed statistical approach for dealing with missing data.

3.3.3 Data Visualisation

Data visualisation is an important skill in applied statistics and machine learning. Statistics does indeed focus on quantitative descriptions and estimations of data. Data visualisation provides an important suite of tools for gaining a qualitative understanding. This can be helpful when exploring and getting to know a dataset and can help with identifying patterns, corrupt data, outliers, and much more. Sometimes data does not make sense until it can look in a visual form, such as with charts and plots.

being able to quickly visualise of data samples and others is an important skill both in applied statistics and in applied machine learning.

3.3.4 Model Prediction

It is important to compare the performance of multiple different machine learning algorithms consistently and it will discover to create a test harness to compare multiple different machine learning algorithms. Each model will have different performance characteristics.

Performance Metrics to calculate

Accuracy: The Proportion of the total number of predictions that is correct otherwise overall how often the model predicts correctly defaulters and non-defaulters.

Precision: The proportion of positive predictions that are actually correct. Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric answer is of all passengers that labelled as survived, how many actually survived? High precision relates to the low false positive rate. We have got 0.788 precision which is pretty good.

F1 Score: It is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution.

The below 2 different algorithms are compared:

1. Linear Support Vector Machine
2. Stochastic Gradient Descent

3.3.5 Model Deployment

In this module the trained machine learning model is converted into hierarchical data format file (.h5 file) which is then deployed in our django framework for providing better user interface and predicting the output based on its categories. Django was designed to be easy to use and extend. The idea behind Django is to build a solid foundation for web applications of different complexity. From then on you are free to plug in any extensions you think you need. It encompasses to build user preference modules.
This means Django provides user with tools, libraries and technologies that allow user to build a web application. This web application can be some web pages, a blog, and a wiki or go as big as a web-based calendar application or a commercial website.

3.3.6 Performance Evaluation

As users interact with the website, send relevant data to the machine learning models for real-time prediction of stress levels. This could be triggered by specific events (e.g., completing a form, spending a certain amount of time on a page). Display the predicted stress levels to users in a user-friendly format, such as a visual indicator or personalized recommendations for stress management techniques.

Additionally the project has a chatbot which allows the users to ask questions to assess and monitor their stress levels. The chatbot offers coping strategies according to the user's stress level.

CHAPTER 4

SYSTEM SPECIFICATIONS

4.1 HARDWARE SPECIFICATIONS

The most common set of requirements defined by any software application is the hardware. The hardware requirements of the system are :

- **Processor :** Inteli3orbetter
- **Memory :** 8GB RAM and 32bit processor

4.2 SOFTWARE SPECIFICATIONS

Software requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. The Software Requirements of the system are:

- **Operating System**

  Windows 10 or later

  Programming language - Python 3.1
- **IDE :** Jupyter Notebook

Tools and Libraries

- Jupyter Notebook
- Visualization Library- Seaborn and Matplotlib
- Data storage and manipulation- csv and pandas libraries
- Deployment- Django

4.2.1 Anaconda Navigator

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows to launch applications and easily manage conda packages, environments, and channels without using command- line commands. Navigator can search for packages on Anaconda.org or in a local Anaconda Repository. Anaconda is created by Continuum Analytics, and it is a Python distribution that comes preinstalled with lots of useful python libraries for data science. It is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. Anaconda comes with many built-in packages that can easily find with conda list on anaconda prompt. As it has lots of packages (many of which are rarely used), it requires lots of space and time as well. If Navigator is an easy, point-and-click way to work with packages and environments without needing to type conda commands in a terminal window. It can be used it to find the packages, install them in an environment, run the packages, and update them – all inside Navigator.

4.2.2 Visual Studio

Visual Studio Code (VS Code) is a lightweight, open-source code editor developed by Microsoft. It is designed to be highly customizable and efficient for developers working on various programming languages and platforms. Here is an overview of how VS Code works:

1. User Interface: VS Code features a clean and intuitive user interface that allows developers to easily navigate and manage their code projects. The

editor includes a sidebar for file navigation, a central editing area, and a terminal for running commands.

2. Extensions: One of the key features of VS Code is its extensive library of extensions. Developers can install extensions to add new functionalities, language support, themes, and tools to customize their coding environment according to their preferences.

3. IntelliSense: VS Code provides IntelliSense, a feature that offers intelligent code completion, suggestions, and auto-completion based on the context of the code being written. This helps developers write code faster and with fewer errors.

4. Debugging: VS Code includes built-in debugging tools that allow developers to set breakpoints, inspect variables, and step through code to identify and fix issues in their applications. The debugging feature supports multiple programming languages and frameworks.

5. Version Control: VS Code integrates with version control systems like Git, enabling developers to manage code repositories, track changes, and collaborate with team members seamlessly within the editor.

6. Customization: VS Code is highly customizable, allowing developers to personalize their coding experience by adjusting settings, installing themes, and configuring keybindings to suit their workflow.

Overall, Visual Studio Code provides a powerful and versatile code editing environment for developers, with a focus on productivity, flexibility, and ease of use.

4.2.3 Django Framework

Django is a micro web framework written in Python. It is classified as a micro-framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Django supports extensions that can add application features as if they were implemented in Django itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. Django was designed to be easy to use and extend. The idea behind Django is to build a solid foundation for web applications of different complexity. Also it is free to build users own modules. Django is great for all kinds of projects. It's especially good for prototyping. Overview of Python Django Framework Web apps are developed to generate content based on retrieved data that changes based on a user's interaction with the site. The server is responsible for querying, retrieving, and updating data. This makes web applications to be slower and more complicated to deploy than static websites for simple applications. Django is considered to be more popular because it provides many out of box features and reduces time to build complex applications. This means Django provides tools, libraries and technologies that allows to build a web application. This web application can be some web pages, a blog, and a wiki or go as big as a web-based calendar application or a commercial website.

4.2.4 Jupyter Notebook

Jupyter Notebook is a versatile and user-friendly integrated development environment (IDE) widely used in research, data science, and programming. It works on Windows, macOS, and Linux and is known for its interactive features. It can extend its functionality with various add-ons and extensions to suit specific needs, including language support and data visualization integrations. What makes Jupyter Notebook stand out is its interactivity and intelligence. It allows users to create documents (notebooks) that combine live code, visualizations, and narrative text, promoting dynamic and collaborative data analysis. The IDE offers smart features like code completion and suggestions, enhancing coding efficiency. It supports multiple programming languages, including Python, R, and Julia. An important advantage is that Jupyter Notebook is open-source, so it's free to use. Its accessibility and extensibility have made it a popular choice among data scientists and researchers.Jupyter notebooks can be converted to an open standard format such as HTML LaTeX, PDF, Markdown, and Python by using the "Download As" function in the web interface.

4.2.5 Matplotlib and Seaborn

Matplotlib is a powerful plotting library in Python used for creating static, animated, and interactive visualizations. Matplotlib's primary purpose is to provide users with the tools and functionality to represent data graphically, making it easier to analyze and understand. Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It is built on top matplotlib library and is also closely integrated with the data structures from pandas. While, Seaborn aims to make visualization the central part of exploring and understanding data. It provides dataset-oriented APIs so that we can switch between different visual representations for the same variables for a better understanding of the dataset. The combination of seaborn's high-level interface and matplotlib's deep customizability will allow both to quickly explore the data and to create graphics that can be tailored into a publication quality final product.

CHAPTER 5

IMPLEMENTATION OF MODULES

5.1 DATASET

The dataset used in this proposed work consists of the following inputs namely Snoring Rate, Respiration Rate, Body Temperature, Limb Movement, Blood Oxygen Level, Retina Eye Movement, Sleeping Rate and Heart Rate to finally determine the Stress Level. The dataset was downloaded from kaggle.com and has about 12000 values.

Figure 5.1 shows the downloaded dataset and the inputs such as Snoring Rate, Respiration Rate, Body Temperature, Limb Movement, Blood Oxygen Level, Retina Eye Movement, Sleeping Rate and Heart Rate to predict the Stress Level.

N1

	A	B	C	D	E	F	G	H	I
1	snoring rate	respiration rate	body temperature	limb movement	blood oxygen	retina eye movement	sleeping rate	heart rate	stress leve
2	93.8	25.68	91.84	16.6	89.84	99.6	1.84	74.2	3
3	91.64	25.104	91.552	15.88	89.552	98.88	1.552	72.76	3
4	60	20	96	10	95	85	7	60	1
5	85.76	23.536	90.768	13.92	88.768	96.92	0.768	68.84	3
6	48.12	17.248	97.872	6.496	96.248	72.48	8.248	53.12	0
7	56.88	19.376	95.376	9.376	94.064	83.44	6.376	58.44	1
8	47	16.8	97.2	5.6	95.8	68	7.8	52	0
9	50	18	99	8	97	80	9	55	0
10	45.28	16.112	96.168	4.224	95.112	61.12	7.112	50.28	0
11	55.52	19.104	95.104	9.104	93.656	82.76	6.104	57.76	1
12	73.44	21.344	93.344	11.344	91.344	91.72	4.016	63.36	2
13	59.28	19.856	95.856	9.856	94.784	84.64	6.856	59.64	1
14	48.6	17.44	98.16	6.88	96.44	74.4	8.44	53.6	0
15	96.288	26.288	85.36	17.144	82.432	100.36	0	75.72	4
16	87.8	24.08	91.04	14.6	89.04	97.6	1.04	70.2	3
17	52.32	18.464	94.464	8.464	92.696	81.16	5.464	56.16	1
18	52.64	18.528	94.528	8.528	92.792	81.32	5.528	56.32	1
19	86.24	23.664	90.832	14.08	88.832	97.08	0.832	69.16	3
20	81.56	22.416	90.208	12.52	88.208	95.52	0.208	66.04	3

STRESS

Figure 5.1 Snapshot of Dataset

5.2 DATA PREPROCESSING

Input: Raw dataset from Kaggle

Output: Processed and cleaned data

Description:

This module focuses on cleaning the raw data ie., To find the missing values, duplicate values, and description of the data type whether it is a float variable or integer. This process is essential because the quality of the data used in machine learning significantly impacts the performance of the models.

Once the consistent data is obtained after the cleaning process, preprocessing of the data is done. Preprocessing involves transforming and preparing the data for analysis.

5.3 DATA VISUALIZATION AND ANALYSIS

Input: Preprocessed data

Output: Data visualization

Description:

This module focuses on visualizing the processed data for deeper analysis and easy interpretation of the values. Data visualization is the representation of data through the use of common graphics, such as charts, plots, infographics, etc.

This project uses various visualization techniques such as count plot, histogram, box plot, violin plot, density graph, heat maps, pie chart, etc. These visual displays of information communicate complex data relationships and data-driven insights in a way that is easy to understand.

Figure 5.3.2 shows the Visual Analysis of the Dataset after Preprocessing. The below figure shows the preprocessed dataset analyzed by bar graphs and histograms where the x axis represents the stress level and the y axis represents the count.

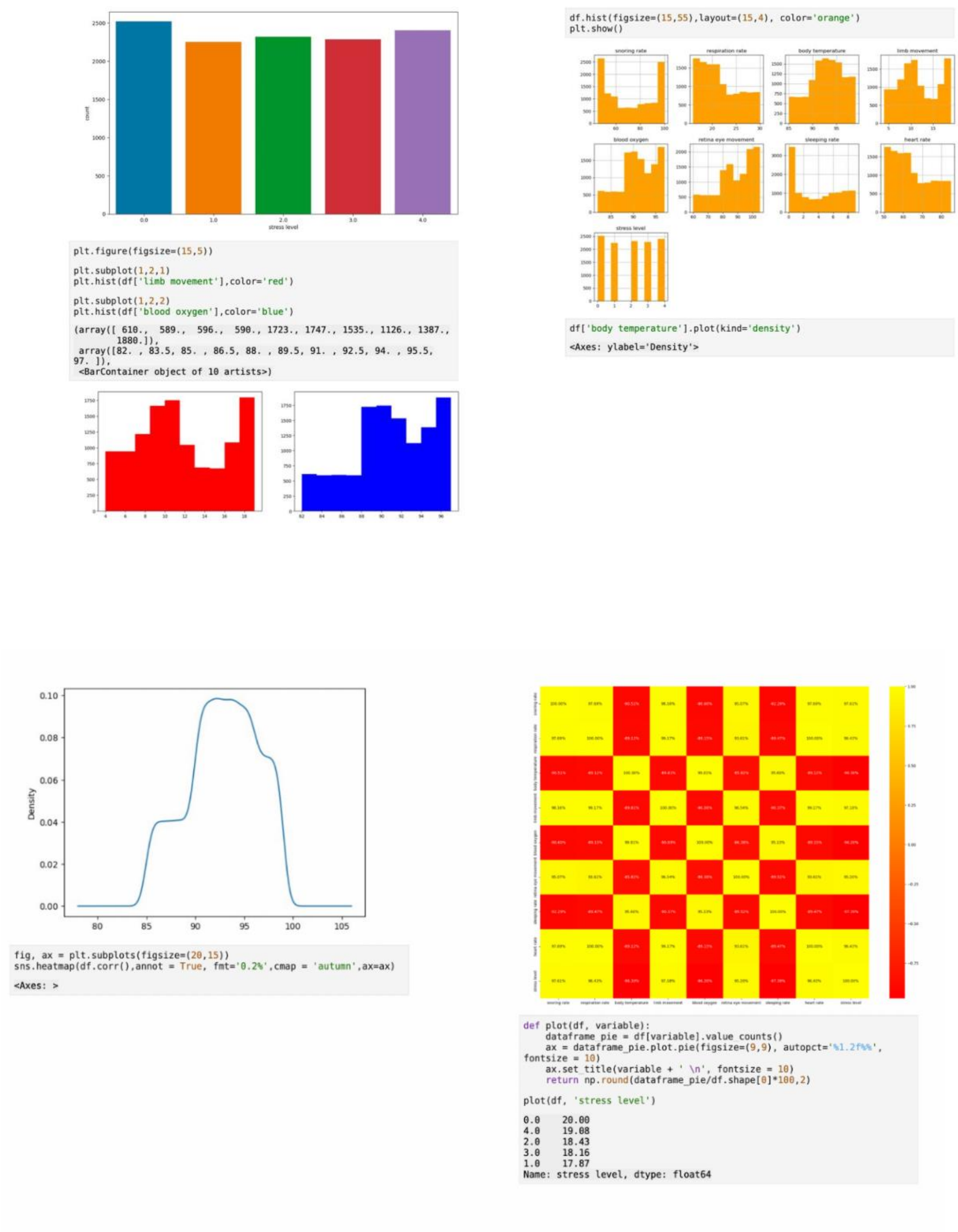


Figure 5.3 Visualization Graphs

5.4 ALGORITHM IMPLEMENTATION

Algorithm 1: Stochastic Gradient Descent

Input: Train and Test data

Output: Predictive model with high accuracy, precision, f1 score.

Description:

Stochastic Gradient Descent (SGD) is a popular optimization algorithm commonly used in machine learning for training models, particularly in the context of large datasets and deep learning neural networks. SGD works by iteratively updating the model parameters in the direction that minimizes the loss function. Instead of computing the gradient of the entire dataset (as in batch gradient descent), SGD computes the gradient of the loss function with respect to a single data point or a small subset of data points (mini-batch) at each iteration. This stochastic nature of SGD allows for faster convergence and makes it well-suited for large-scale optimization problems. During each iteration, the model parameters are updated using the gradient of the loss function with respect to the current data point or mini-batch, scaled by a learning rate. The learning rate determines the size of the step taken in the parameter space and influences the convergence speed and stability of the optimization process. SGD is known for its efficiency and scalability, making it a popular choice for training machine learning models, especially in scenarios where computational resources are limited or when dealing with large datasets. However, SGD can be sensitive to the choice of hyperparameters, such as the learning rate and mini-batch size, and may require careful tuning to achieve optimal performance.

Algorithm 2: Linear Support Vector Machine

Input: Train and Test data

Output: Predictive model with accuracy, precision, f1 score.

Description:

Linear Support Vector Machine is a supervised machine learning algorithm used for classification and regression tasks. It works by finding the optimal hyperplane that best separates the data points into different classes in a high-dimensional space. The hyperplane is chosen in such a way that it maximizes the margin, which is the distance between the hyperplane and the nearest data points from each class, known as support vectors.

1. Data representation: The input data is represented as feature vectors in a high-dimensional space, where each feature corresponds to a different aspect of the data (e.g., physiological signals, behavioral patterns).
2. Hyperplane optimization: The Linear SVM algorithm aims to find the hyperplane that maximizes the margin between different classes of data points. The margin is the distance between the hyperplane and the nearest data points from each class.
3. Classification: Once the optimal hyperplane is determined, the Linear SVM can classify new data points by assigning them to one of the classes based on which side of the hyperplane they fall on.
4. Margin and support vectors: The data points that lie closest to the hyperplane are known as support vectors. The margin is maximized by ensuring that the support vectors are correctly classified and are at the maximum distance from the hyperplane.
5. Regularization: In the case of Linear SVM, regularization parameters can be used to control the trade-off between maximizing the margin and minimizing classification errors, thus preventing overfitting.

5.5 DEPLOYMENT USING DJANGO

Input: Medical values

Output: Stress level result

Description:

This module focuses on the application aspect of the proposed work, utilizing Django for deployment. This process involves users inputting stress prediction values, including parameters such as Snoring rate, Respiration rate, Body Temperature, Limb Movement, Blood oxygen levels, and more. Upon entering these values, the system utilizes the predictive model to determine the stress level, categorizing it as Very Low, Low, Medium, High, or Very High. The predicted stress level is then displayed as the result, providing users with valuable insights into their current stress levels.

This module streamlines the process of stress assessment and management, offering a user-friendly interface for individuals to monitor and address their stress levels effectively. By leveraging Django for deployment, it ensures a seamless and efficient experience for users seeking to understand and improve their mental well-being.

Additionally, the project has a chatbot which allows the users to ask questions to assess and monitor their stress levels. The chatbot offers coping strategies according to the users stress level. The chatbot serves as an emotional support that can provide emphatetic responses, encouragement. Overall, a stress level chatbot aims to provide users with a supportive and accessible tool for monitoring and addressing stress, promoting mental well-being, and fostering self-care practices.

CHAPTER 6

RESULTS AND DISCUSSION

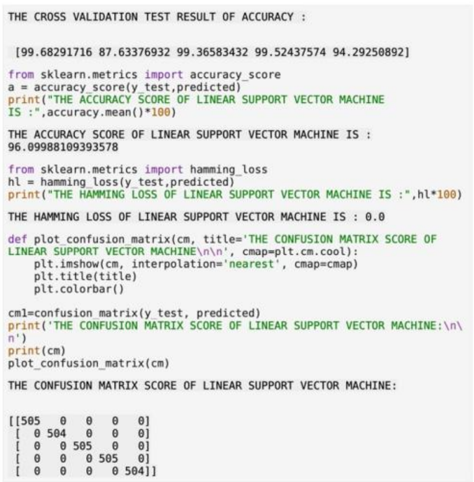
The results of our stress level prediction using Linear SVM and Gradient Boosting techniques have shown promising outcomes.

The Linear Support Vector Machine (SVM) model demonstrated strong performance in accurately predicting stress levels based on the input features. The model effectively separated the data points into different stress level categories, achieving a high level of precision and recall. This indicates that the Linear SVM algorithm is well-suited for classifying stress levels and can provide reliable predictions for individuals.

Stochastic Gradient Descent (SGD) is a popular optimization algorithm commonly used in machine learning for training models, particularly in the context of large datasets and deep learning neural networks. SGD works by iteratively updating the model parameters in the direction that minimizes the loss function. Instead of computing the gradient of the entire dataset (as in batch gradient descent), SGD computes the gradient of the loss function with respect to a single data point or a small subset of data points (mini-batch) at each iteration. This stochastic nature of SGD allows for faster convergence and makes it well-suited for large-scale optimization problems.

The comparison between Linear SVM and Stochastic Gradient Descent (SGD) techniques revealed that both models have their strengths in predicting stress levels. The choice of algorithm may depend on the specific requirements of the application, such as the need for interpretability, computational efficiency, or predictive accuracy. Overall, the results and discussion highlight the effectiveness of Linear SVM and Stochastic Gradient Descent(SGD) techniques in predicting stress levels, offering valuable insights for developing accurate and reliable tools for stress management and well-being.

Figure 6.1 shows the implementation of Linear Support Vector Machine Algorithm. The below figure shows the number of train dataset and test dataset. It also shows the confusion matrix score of the LSVM Classifier and the Accuracy along with Hamming Loss.



```
THE CROSS VALIDATION TEST RESULT OF ACCURACY :

 [99.68291716 87.63376932 99.36583432 99.52437574 94.29250892]

from sklearn.metrics import accuracy_score
a = accuracy_score(y_test,predicted)
print("THE ACCURACY SCORE OF LINEAR SUPPORT VECTOR MACHINE
IS :",accuracy.mean()*100)

THE ACCURACY SCORE OF LINEAR SUPPORT VECTOR MACHINE IS :
96.09988109393578

from sklearn.metrics import hamming_loss
hl = hamming_loss(y_test,predicted)
print("THE HAMMING LOSS OF LINEAR SUPPORT VECTOR MACHINE IS :",hl*100)

THE HAMMING LOSS OF LINEAR SUPPORT VECTOR MACHINE IS : 0.0

def plot_confusion_matrix(cm, title='THE CONFUSION MATRIX SCORE OF
LINEAR SUPPORT VECTOR MACHINE\n\n', cmap=plt.cm.cool):
    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()

cm1=confusion_matrix(y_test, predicted)
print('THE CONFUSION MATRIX SCORE OF LINEAR SUPPORT VECTOR MACHINE:\n\
n')
print(cm)
plot_confusion_matrix(cm)

THE CONFUSION MATRIX SCORE OF LINEAR SUPPORT VECTOR MACHINE:

[[505   0   0   0   0]
 [  0 504   0   0   0]
 [  0   0 505   0   0]
 [  0   0   0 505   0]
 [  0   0   0   0 504]]
```

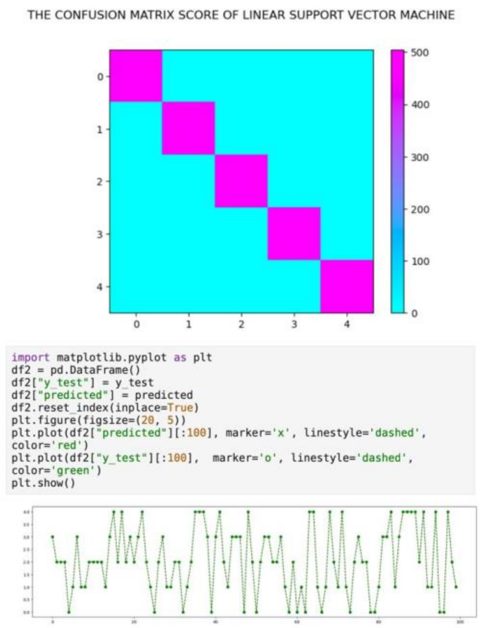


Figure 6.1 Implementation Snapshot of LSVM Algorithm

Figure 6.2 shows the implementation of SGD Algorithm. The below figure shows the number of train dataset and test dataset. It also shows the confusion matrix score of the SGD Classifier and the Accuracy along with Hamming Loss.

Figure 6.2 Implementation Snapshot of SGD Algorithm

Figure 6.3 shows the Landing Page of the User Interface created for the proposed work which redirects the user to the Login Page and the Registration Page when the user clicks Explore.

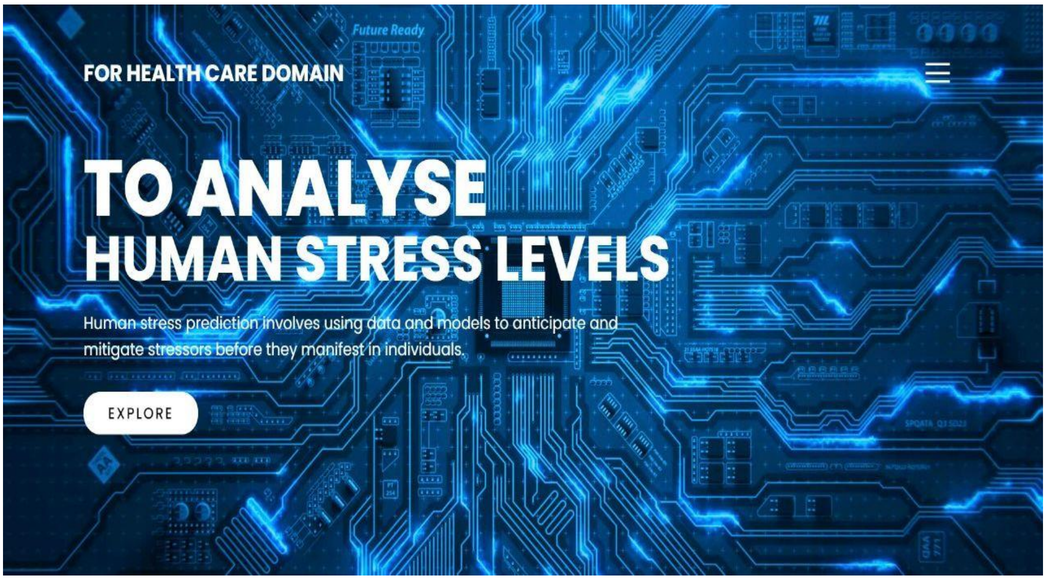


Figure 6.3 Landing Page

Figure 6.4 shows the registration page which consists of username, email, password and password confirmation for the new users to sign in to the website.

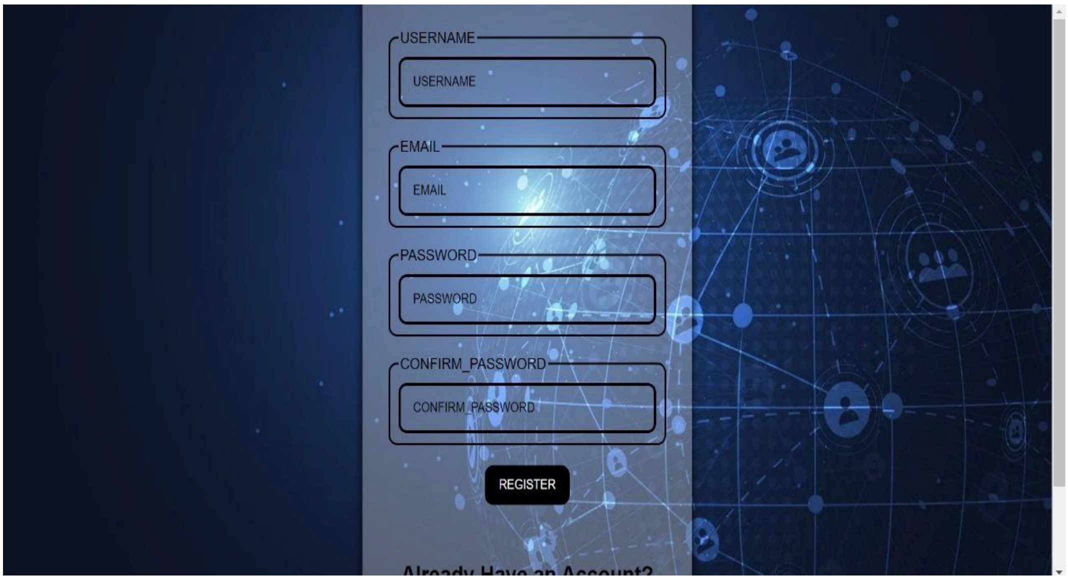


Figure 6.4 Registration Page

Figure 6.5 shows the Login Page for the users which consists of username and password for the existing users to login to the website.

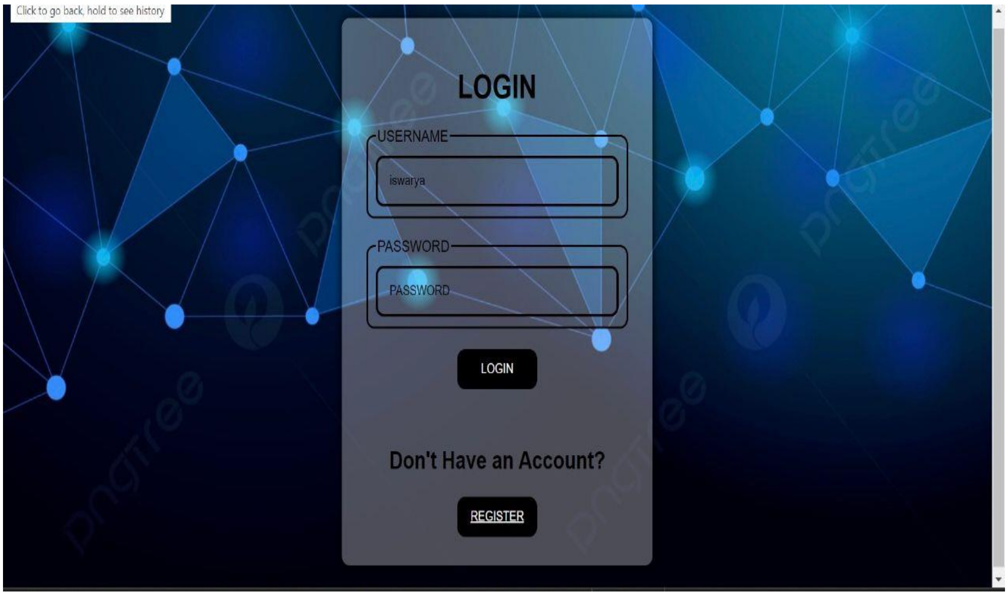


Figure 6.5 Login Page

Figure 6.6 shows the Home Page of the website which redirects to the preferred page of the user along with information about Preventions from Mental Stress and Precautions to Mental Stress.

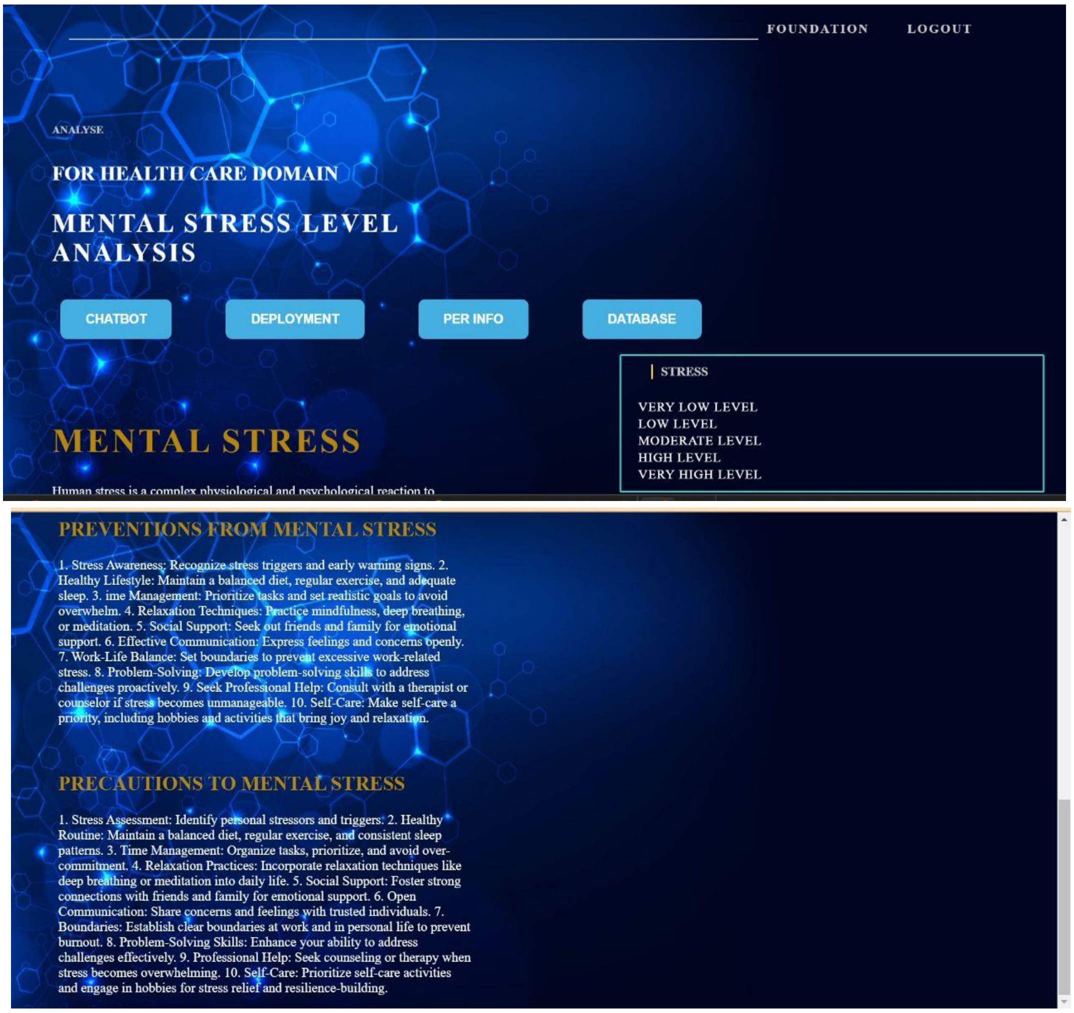


Figure 6.6 Home Page

Figure 6.7 shows the Stress Prediction page. The user enters the input for Snoring Rate, Respiration Rate, Body Temperature, Limb Movement, Blood Oxygen Level, Retina Eye Movement, Sleeping Rate and Heart Rate and clicks Predict to predict the Stress Level.

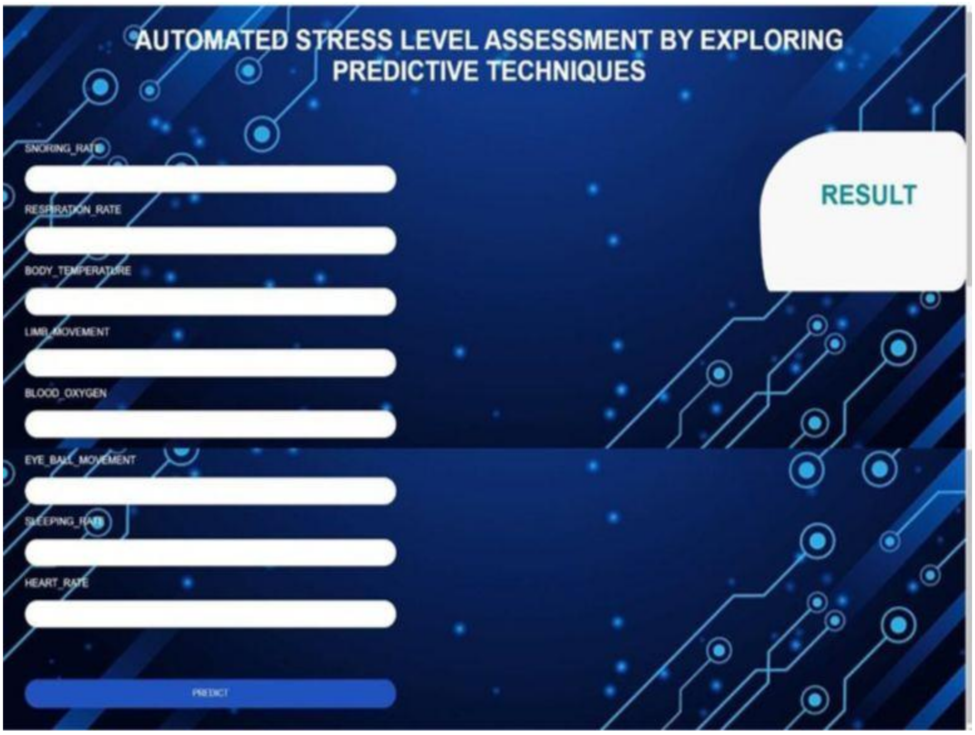


Figure 6.7 Stress Prediction Page

Figure 6.8 The below screenshot shows the Stress Level being predicted for inputs corresponding to Level 0 which is Very Less Depression along with its preventive measures.

Figure 6.8 Snapshot of Stress Level 0

Figure 6.9 The below screenshot shows the Stress Level being predicted for inputs corresponding to Level 1 which is Less Depression along with its preventive measures.

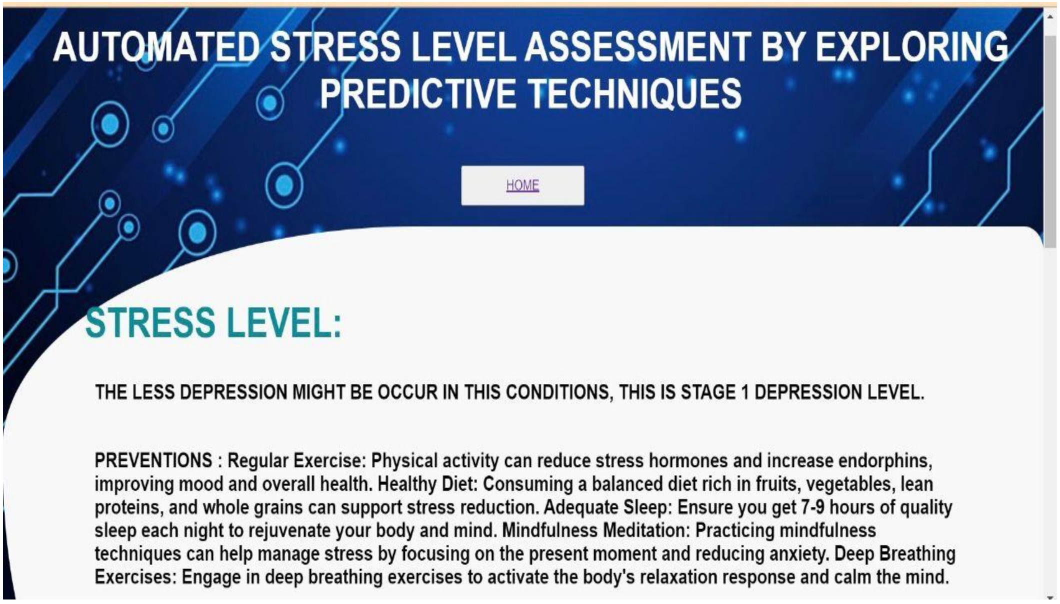


Figure 6.9 Snapshot of Stress Level 1

Figure 6.10 The below screenshot shows the Stress Level being predicted for inputs corresponding to Level 3, which is High Depression along with its preventive measures.

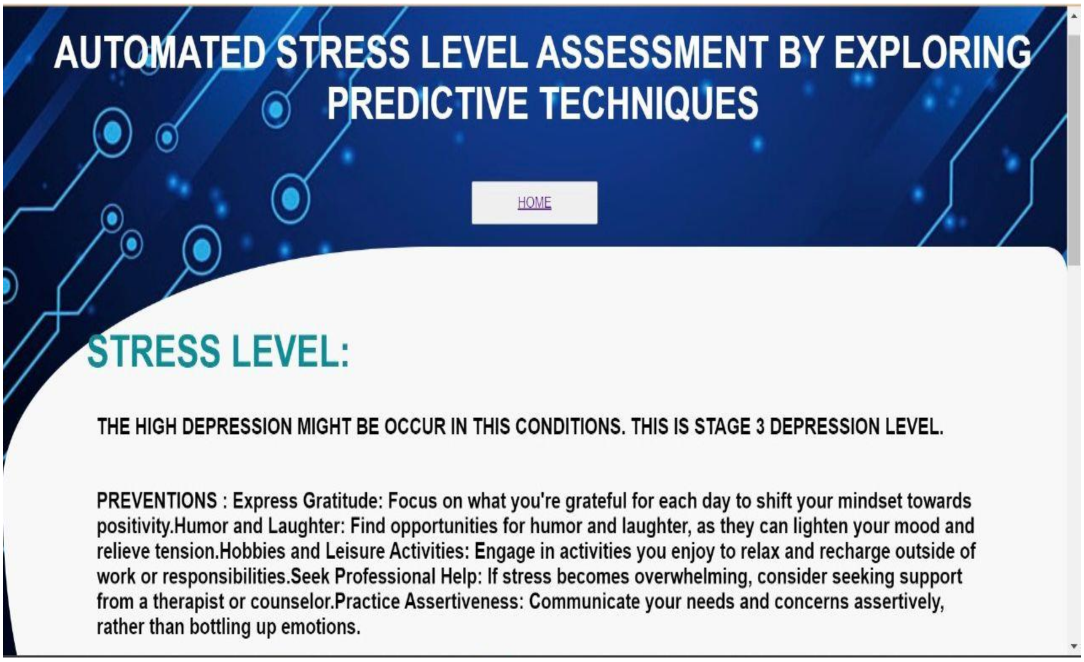


Figure 6.10 Snapshot of Stress Level 3

Figure 6.11 The below screenshot shows the Stress Level being predicted for inputs corresponding to Level 4, which is Very High Depression along with its preventive measures.

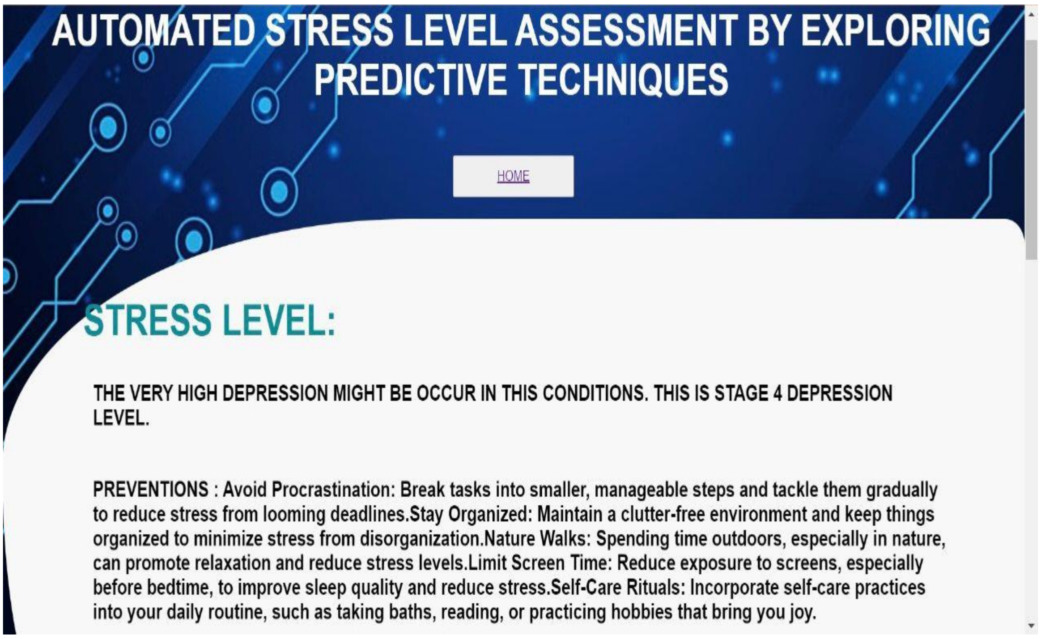


Figure 6.11 Snapshot of Stress Level 4

Figure 6.12 shows the Chatbot Page where the user enters their queries and questions. User can ask stress related questions to the chatbot where, it returns the suggested answer to the user.

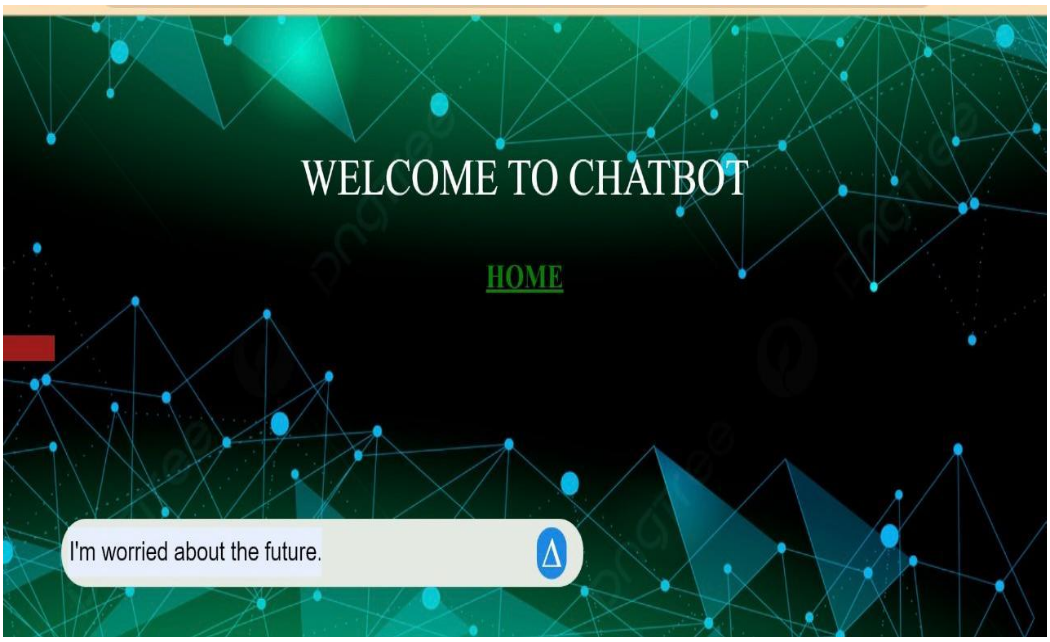


Figure 6.12 Snapshot of Chatbot Question

Figure 6.13 shows the Chatbot Page where the query is being answered. The below screenshot shows the chatbot answering to the question the user have asked.

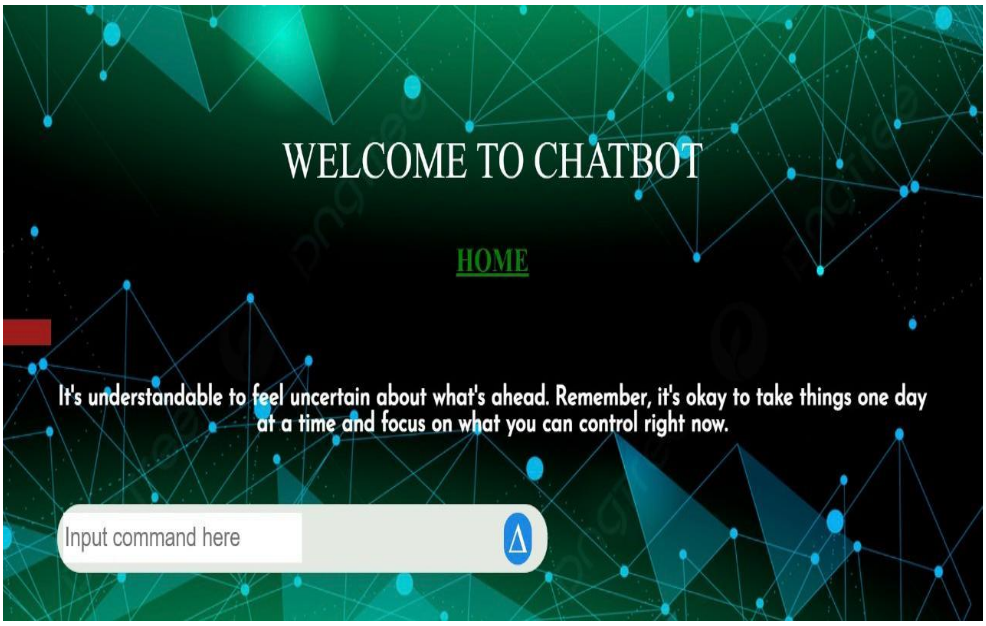


Figure 6.13 Snapshot of Chatbot Answer

CHAPTER 7

CONCLUSION AND FUTURE WORK

In conclusion, our project leveraging predictive techniques and machine learning algorithms to predict stress levels represents a significant step forward in the field of mental health assessment and management. By developing a robust predictive model that considers various factors contributing to stress, it has the potential to revolutionize how individuals monitor and address their stress levels. Through personalized recommendations and interventions, the proposed work aims to empower individuals to take proactive steps towards managing their stress and improving their overall well-being. Moving forward, this project committed to further refining the predictive model, collaborating with experts in the field, and ultimately making a positive impact on mental health outcomes.

Deploying the project in the cloud offers numerous advantages for optimizing the implementation of stress level prediction. By leveraging cloud infrastructure, we can achieve scalability, flexibility, and cost-effectiveness in deploying our predictive model for stress levels. The cloud environment provides the necessary resources to handle large datasets and complex machine learning algorithms efficiently. Overall, deploying the stress level prediction project in the cloud and optimizing it for implementation in an IoT system enhances the efficiency, effectiveness, and reach of the predictive model, ultimately contributing to improved mental health outcomes and well-being for individuals.

REFERENCES

1. Zhang, Xiaowei, et al. (2023) "Dynamic alignment and fusion of multimodal physiological patterns for stress recognition." IEEE Transactions on Affective Computing. 1-12.

2. Magtibay, Karl, and Karthikeyan Umapathy. " (2023) A Review of Tools and Methods for Detection, Analysis, and Prediction of Allostatic Load due to Workplace Stress." IEEE Transactions on Affective Computing, 1-22.

3. Dennis Rodrigo, Zelun Wang, and Ricardo Gutierrez-Osuna. (2021) "Towards participant-independen stress detection using instrumented peripherals." IEEE Transactions on Affective Computing.

4. Yasmin, Hena, Salman Khalil, and Ramsha Mazhar. (2020) "COVID 19: Stress management among students and its impact on their effective learning." International technology and education journal 4.2, 65-74.

5. Chouhan, Devraj Singh. (2016) "Stress and Its Major Effects on Human Health." International Journal of Multidisciplinary Allied Research Review and Practices.

6. Shahsavarani, et al. (2015) "Stress: Facts and theories through literature review." International Journal of Medical Reviews 2.2, 230-241.

7. Smith, Melinda, Ellen Jaffe-Gill, and Robert Segal. "Stress Management: How to Reduce, Prevent, and Cope with Stress." Helpguide.org: Understand, Prevent and Resolve Life's Challenges. Helpguide.org, Dec. 2008. Web. 20 Apr. 2010.

8. Shahsavarani AM, Ashayeri H, Lotfian M, Sattari K. The effects of Stress on Visual Selective Attention: The Moderating Role of Personality Factors. Journal of American Science. 2013; 9(6s): 1-16.

9. J. Diemer, N. Lohkamp, A. Muhlberger, and P. Zwanzger, "Fear and physiological arousal during a virtual height challenge effects in patients with acrophobia and healthy controls," J. Anxiety Dis- orders, vol. 37, pp. 30–39, 2016.

10. Roche M, Haar J. Aspirations and job burnout: A study of New Zealand managers. In: Community, Work and Family 71. IV International Conference, 19–21 May 2011, Tampere, Finland; 2011.

11. K. Debattista, T. Bashford-Rogers , C. Harvey, B. Waterfield, and A.

Chalmers, “Subjective evaluation of high-fidelity virtual envi- ronments for driving simulations,” IEEE Trans. Hum. -Mach. Syst., vol. 48, no. 1, pp. 30–40, Feb. 2018.

12. A.-M. Brouwer and M. A. Hogervorst, “A new paradigm to induce mental stress: The sing-a-song stress test (SSST),” Front. Neurosci., vol. 8, pp. 224–232, 2014.

13. L. Johnson, M. Hou, E. Prager, and J. LeDoux , “Regulation of the fear network by mediators of stress: Norepinephrine alters the balance between cortical and subcortical afferent excitation of the lateral amygdala,” Front. Behav. Neurosci., vol. 5, pp. 23–30, 2011.

14. D. Gromer, M. Reinke, I. Christner, and P. Pauli, “Causal interac- tive links between presence and fear in virtual reality height exposure,” Front. Psychol., vol. 10, pp. 141–152, Jan. 2019.

15. A. Alberdi, A. Aztiria, and A. Basarab, “Towards an automatic early stress recognition system for office environments based on multimodal measurements: A review,” J. Biomed. Inform., vol. 59, pp. 49–75, 2016.

16. D. McDuff, S. Gontarek, and R. Picard, “Remote measurement of cognitive stress via heart rate variability,” in Proc. 36th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc., 2014, pp. 2957–2960.

17. S.-H. Lau, “Stress detection for keystroke dynamics,” Ph.D disser-tation, Machine Learning Department School of Computer Sci-ence, Carnegie Mellon Univ., Pittsburgh, PA, USA, 2018.

18. Almojali, A. I., Almalki, S. A., Alothman, A. S., Masuadi, E. M., & Alaqeel, M. K. (2017). The prevalence and association of stress with sleep quality among medical students. Journal of Epidemiology and Global Health, 73, 169–174.

19. Bayram, N., & Bilgel, N. (2019). The prevalence and socio-demographic correlations of depression, anxiety and stress among a group of university students. Social Psychiatry and Psychiatric Epidemiology, 438, 667–672.

20. Brooks SK, Webster RK, Smith LE, Woodland L, Wessely S, Greenberg N, et al.(2020). The psychological impact of quarantine and how to reduce it: rapid review of the evidence. Lancet. 395(10227):912–20.

21. Eisenberg, D., Gollust, S. E., Golberstein, E., & Hefner, J. L. (2007). Prevalence and correlates of depression, anxiety, and suicidality among university students. The American Journal of Orthopsychiatry, 774, 534–542.